GET RICH OR DIE TRYING… UNPACKING REVENUE MODEL CHOICE USING MACHINE LEARNING AND MULTIPLE CASES

RON TIDHAR

Department of Management Science and Engineering
Stanford University
rtidhar@stanford.edu

KATHLEEN M. EISENHARDT

Department of Management Science and Engineering
Stanford University
kme@stanford.edu

December 7, 2018

Revise and resubmit, Strategic Management Journal
Special Issue on Question-Driven Research

ABSTRACT

While revenue models are gaining strategic importance, related research is incomplete. Thus, we ask the phenomenon-driven question: “When should particular revenue models be used?” We use a novel theory-building method that blends exploratory data analysis, machine learning, and multi-case theory building. Our sample is from the AppStore, an economically important setting in the digital economy. Our primary contribution is a framework of configurations of effective revenue models. It indicates new theoretical relationships linking quality signals, user resources, and product complexity to the choice of revenue model. It also unlocks equifinal paths and new revenue models (e.g., bundled and fragmented). Overall, we contribute a more accurate and theoretical view of effective revenue models. We also highlight the surprising complementarity of machine learning and multi-case theory building.

Keywords:
Revenue models, competition, mobile application products (apps), machine learning, multi-case theory building.
INTRODUCTION

In 2011, music streaming service, Spotify, launched its product to U.S. listeners. Although similar to other music streaming services, Spotify was notably different in its revenue model. While other companies relied on either a paid model (e.g., Apple iTunes) or an advertising one (e.g., Pandora), Spotify offered its service for free, but included a paid premium version. The premium product provided more content, greater functionality, and a better user experience. The freemium revenue model (i.e., free up-front, with premium up-sells) helped Spotify to double its revenue to over $500 million in a year and become a runaway success (Statista, 2018a). While Spotify's freemium revenue model is now well-established, it was unclear at the outset which revenue model to choose.

As the Spotify vignette suggests, effective revenue models are essential for firm success. Consistent with others (Casadesus-Masanell and Zhu, 2010; Johnson et al., 2008), we define a revenue model as the monetization approach by which a firm earns revenues from the sales of its products and services. Thus, revenue models are the means by which firms capture value, and earn the revenue that is essential for superior financial performance. Prior research suggests that executives such as those at Spotify understand that designing a revenue model (and a related business model) is an early and critical choice (McDonald and Eisenhardt, 2018).1

Revenue models are important. First, revenue models are a novel source of innovation (Casadesus-Masanell and Zhu, 2013; Gilbert, 2005; Snihur and Zott, 2014). While some firms focus on traditional innovations like new products, new paths to value creation (business models) and value capture (revenue models) can enable disruptive ways of competing (Massa et al., 2017; Snihur and Zott, 2014).

---

1 Consistent with prior research (e.g., Zott and Amit, 2010; Massa et al., 2017), we separate business models and their focus on value creation from revenue models and their focus on value capture. We appreciate a reviewer’s comment here.
Zott et al., 2011). Second, blurred industry lines, lower barriers to entry, and Internet penetration have created opportunities for complementors and users (not just producer-firms) to create value, thus enabling more varied and novel revenue models to capture it (Hannah and Eisenhardt, 2018; Gambardella and McGahan, 2010). Indeed, figuring out a revenue model is increasingly a primary strategic challenge (Massa et al., 2017), and reshaping revenue models (and restructuring related activity systems) are significant avenues for success (Gilbert, 2005; Kim and Min, 2015) and even survival (Seamans and Zhu, 2017).

Yet while revenue models are important for innovation and performance, it is not always obvious which to choose. For example, McDonald and Eisenhardt (2018) study five ventures in the social investing market. They find that some teams chose an advertising revenue model that was ultimately low-performing. By contrast, other teams developed a paid revenue model that relied on transaction fees that proved to be successful. In addition, while prior work recognizes many types of revenue models, their importance remains elusive. On the one hand, practitioner work often favors long atheoretical lists that may be too many to be meaningful (e.g., Gassmann, Frankenberger et al., 2014 indicate 55). On the other hand, academic work often pares the list to only a few: (1) paid, (2) advertising, and (3) freemium (e.g., Arora et al., 2017; Liu et al., 2012; Rietveld, 2018). Yet this may be too few for capturing essential differences.

While prior research suggests broad factors (e.g., product quality, consumer ad-aversion, and advertising rates) that may shape the optimal choice of revenue model, this research is often incomplete (e.g., glosses over the advertiser perspective), too simple (e.g., under-theorizes freemium models), and conflicting (e.g., inconsistent results for product quality). As a result, this work leaves open which are the most relevant revenue models, why similar products (e.g.,

---

2 While helpful and well-known, Gassmann et al., 2014 mix and overlap freemium models, miss fragmented v. bundled distinction, overlap models, and sometimes use the same examples to illustrate different models.
Pandora and Spotify, StitchFix and Rent the Runway, Netflix and Hulu, 23andMe and Pathway Genomics) have different revenue models, and when particular revenue models are likely to be effective. Overall, theoretical insights into an important empirical puzzle – when and why is each revenue model is preferred – remain limited. We address this gap with question-driven research. We ask, “When should particular revenue models be used?”

We tackle our research question by studying 66,652 mobile products (“apps”) on Apple’s App Store. This setting is appropriate because it is economically significant (Arora et al., 2017; Askalidis, 2015; Ifrach and Johari, 2014), provides accurate measures of relevant constructs (Yin et al., 2014), and covers a wide range of products in multiple industries (Bresnahan et al., 2014; Hallen et al., 2018), likely yielding a rich understanding of revenue models in different settings. Consistent with question-driven research, we use a novel theory-building methodology: (1) **Exploratory data analysis** to reveal broad trends and to provide a roadmap for the theoretical sampling of cases. (2) **Multi-case theory building** to identify key constructs and theoretical mechanisms, and to seed machine learning. (3) **Machine learning** to provide large-scale analysis to replicate and extend the theory-building insights from the case studies with more precise non-linearities, equifinal paths, and effect sizes.

We contribute to the strategy literature in several ways. First, using a question-driven approach, we contribute a theoretical framework for effective revenue models (Table 1). Second, we identify key constructs – i.e., user resources, quality signals, and product complexity – in new theoretical relationships within effective revenue model configurations. These contrast with the traditional emphasis on product quality, ad-aversion, and advertising rates, and offer a more

---

3 We use some known concepts but do so in new theoretical relationships with new underlying arguments and dependent variables. Using known constructs fits with question-driven research such as ours (Special Issue call). We appreciate a reviewer’s raising this issue.
accurate and theoretical view of revenue models. Third, we unlock equifinal paths, and introduce new and important revenue models (e.g., fragmented freemium, bundled freemium). Broadly, we contribute a novel, question-driven approach to building theory that blends machine learning, multi-case theory building and exploratory data analysis. Specifically, we highlight surprising complementarities and similarities between multiple cases and machine learning. To the best of our knowledge, our paper is among the first in strategy to use machine learning to build theory.

BACKGROUND

Prior research examines revenue models primarily with formal modeling and hypothesis-testing methods. It identifies three primary revenue models: (1) paid - product or service is sold at an up-front price, (2) advertising - product is free to the user and revenue is earned from paying advertisers, and (3) freemium revenue model - combines two products – a free, often ad-sponsored version, and a paid, “premium” version. There are several research streams.

Research on single revenue models

One stream provides insight into a single revenue model. Research on the advertising model broadly finds that it is less effective when consumers exhibit high “ad-aversion” (Gabszewicz et al., 2005; Lin et al., 2012). Further work suggests, however, that consumer targeting can mitigate ad-aversion. For example, Iyer and colleagues (2005) construct a formal analytic model of an advertiser that shows that better targeting to a specific consumer segment always increases revenues. Research on the paid revenue model explores quality and consumer awareness. For example, Duan and colleagues (2008) study the effects of these factors on movie ticket sales. The authors find that consumer awareness (measured by number of reviews) has a positive and significant impact on subsequent sales. Surprisingly, however, quality (measured by user ratings) does not. In other words, consumer awareness, rather than quality, drives sales of products (i.e., movies) with a paid revenue model.
Research on two revenue models

A second stream compares two revenue models such as paid v. advertising. For example, Eckhardt (2016) studies applications (apps) in the Palm and Smartphone ecosystems. The author finds that revenue of paid apps increases when there are more advertising-supported apps in the product segment. The author argues that developers of paid apps learn about user preferences from rivals (i.e., user choices of advertising-supported products), and then design higher quality ones. If, however, there are many paid apps, this advantage decreases because greater competition and converging quality drive down revenue. In short, the paid revenue model is more effective for higher quality products – i.e., those that better fit user preferences.

In other work, Chen, Fan, and Li (2016) develop a formal analytic model of e-commerce platforms with a paid or advertising revenue model. In particular, the authors observe that eBay uses a paid model, while Taobao uses an advertising one. Under the paid revenue model, the platform charges sellers a transaction fee for each sale. In contrast, under the advertising model, the platform allows sellers to pay for advertising their offerings. The key insight is that when the platform quality is low (i.e., low likelihood of finding good matches with the platform’s search function), an advertising model is better. The rationale is that advertising can improve sellers’ chances of a match (i.e., a sale), and so overcome the platform’s poor quality.

In short, an advertising revenue model is more effective for lower quality products (Chen et al., 2016) – which complements Eckhardt’s (2016) finding that a paid revenue model is better for higher quality products. Thus, research on advertising v. paid revenue models finds that product quality is a key variable that influences the choice.

A related stream unpacks the choice between paid v. freemium revenue models. Using a formal analytic model of a monopolist, Gabszewicz et al. (2005) find that ad-aversion (i.e., consumer dislike of advertising) shapes revenue model choice. If ad-aversion is high, a
monopolist should choose a paid revenue model (i.e., in order to raise prices to maximize profits). In contrast, if ad-aversion is low, the monopolist can benefit both from advertising and higher prices because consumers may see advertising as valuable. In this case, the monopolist should choose a freemium revenue model.

Rietveld (2018) examines the performance implications of paid v. freemium revenue models for games on Steam, a PC gaming platform. Games with a paid model receive more playing time and earn more revenue than those with a freemium model. The author follows up with a lab experiment of the freemium model in which users are offered multiple up-sell products (v. one). The main result is that more up-sell choices earn more revenue. Thus, the paid revenue model outperforms the freemium revenue model. Within freemium, more up-sell variety increases performance. Relatedly, Arora and colleagues (2017) analyze apps on Google Play Store. They find that the freemium model results in slower adoption of the paid product. They speculate that the free product cannibalizes the sales of the premium product – i.e., users would adopt the paid product if the free version did not exist.

Yet other work contradicts these findings. Liu et al. (2012) study 1,597 app products on the Google Play Market (i.e., Android devices). The authors find that the freemium revenue model outperforms the paid model in terms of downloads and revenue. This effect is stronger in the Games category. The argument is that users particularly value their own experiences for categories like games. Thus, the opportunity to play a game before upgrading (i.e., freemium model) is preferred to the up-front risk of the paid model.

In short, the studies of freemium revenue models are often conflicting. Some work highlights that the freemium revenue model is preferred because it allows users to try the product before paying (Liu et al., 2012). In contrast, other work indicates a preference for the paid model.
because the freemium model may cannibalize the paid premium product, leading to overall lower performance (Arora et al., 2017; Rietveld, 2018).

**Research on three revenue models**

A third stream considers the choice among all three revenue models. For example, Prasad and colleagues (2003) compare the choice of an advertising, paid, or freemium revenue model by a monopolist. Using formal analytic modelling, they find that, except in a narrow range of circumstances, a monopolist should offer the freemium model. That is, it should segment its customer base by offering a free (or nearly free) product supported by advertising and a paid (subscription) premium product. They further find that the balance between advertising and paid depends on product quality. That is, a monopolist with a higher quality product should tip its freemium model in favor of paid revenue. In contrast, a monopolist with a lower quality product should tip in favor of advertising revenue. In short, the freemium model is preferred because it allows customer segmentation. As product quality increases, more of the freemium revenues should come from the paid version.

In other work, Casadesus-Masanell and Zhu (2010) use a formal analytic model to explore the performance of revenue models in a competitive setting. Specifically, they consider a monopolist incumbent’s choice of revenue model when faced with a low quality, ad-sponsored entrant. The authors show that in a monopoly setting, the monopolist prefers the freemium model across a wide range of settings because this allows customer segmentation based on willingness to pay. In contrast to a monopolistic setting, however, the authors conclude that the incumbent should compete with an advertising or a paid (i.e., “pure” revenue model), *not* a freemium (i.e., hybrid one). The rationale is that the freemium model suffers from self-cannibalization, while pure revenue models do not. Further, advertising rate drives the choice between an advertising v. paid revenue model. If the advertising rate is low, the incumbent should choose a paid revenue
model to increase revenue. If the advertising rate is high, the incumbent should choose the advertising model because it allows the incumbent to capture the entire market without self-cannibalization and push out the entrant.

**Summary**

Taken together, the revenue model research offers insights into the types and choices of revenue models. An *advertising* revenue model is more effective with increasing advertising rate (Casadesus-Masanell and Zhu, 2010; Lin et al, 2012) and decreasing ad-aversion (Gabszewicz et al., 2005; Lin et al., 2012; Riggins, 2002). This model is also preferred when product quality is low because consumers are unlikely to pay (Casadesus-Masanell and Zhu, 2010; Chen et al., 2016; Lin et al., 2012). Yet, while these explanations are plausible, they largely neglect the advertiser’s perspective. So, they leave open when and why advertisers would be willing to pay high advertising rates, and why advertisers would want to associate with a low-quality product.

Similarly, prior research offers insights into the choice of a *paid* revenue model. This model is effective when product quality is high (Casadesus-Masanell and Zhu, 2010; Chen et al., 2016; Eckhardt, 2016; Prasad et al., 2003). In contrast, other work finds that consumer awareness (not product quality) drives the effectiveness of a paid revenue model (Duan et al., 2008). A paid revenue model is also effective when the advertising rate is low (Casadesus-Masanell and Zhu, 2010; Gabszewicz et al., 2005; Lin et al., 2012). Yet again, the advertiser’s perspective is missing – i.e., when and why will an advertiser pay more?

Finally, research highlights advantages of *freemium* models (v. paid models) such as customer segmentation (Prasad et al., 2003) and ability to experience the product before paying (Liu et al., 2012). Yet other work finds the freemium model to be inferior (Rietveld, 2018; Casadesus-Masanell and Zhu, 2010; Arora et al., 2017) because self-cannibalization renders it ineffective, especially with competition. Still other work finds few results (e.g., Hamari et al.,
Overall, this lack of clarity suggests that the freemium revenue model is under-theorized, with one or more constructs related to performance not yet recognized.

In sum, prior work is often incomplete (e.g., misses the advertiser perspective), too simple (e.g., under-theorizes freemium models), and conflicting (e.g., inconsistent results for product quality). As such, understanding of an increasingly important strategic puzzle – when and why should each revenue model be used – remains limited. To address this question-driven research gap, we adopt a novel theory-building approach.

METHODS

Research sample and setting

We sampled from the entire population of mobile products (“apps”) available on Apple’s iOS App Store in November, 2015. The App Store is a “store front” through which consumers can browse, review, and download apps for their iOS devices. These apps span 22 market categories, including Travel, Social Networking, and Games. Launched in 2008, the App Store has garnered significant attention, investment, and success. To date, there have been over 2.2 million products developed for the iOS App Store (Statista, 2018c), generating over $70 billion in cumulative earnings for developers (Apple, 2017).

The App Store is an attractive setting for our research for several reasons. First, it contains a wide variety of applications products (apps) that span diverse products and markets, ranging from finance and medical products to games and travel. This provides a wide swath of products and market categories in which to observe revenue models. Second, the App Store (and other such stores) are economically important and central to the digital economy (e.g., $143B in revenues and 12 million developers in 2016) (Arora et al., 2017; Yin et al., 2014). Third, the App Store provides accurate measures of key constructs related to our research question including revenue model, market category, and performance. Indeed, the growing number of studies across
a range of topics using these data attests to their accuracy and value (e.g. Askalidis, 2015; Bresnahan et al., 2014; Hallen et al., 2018; Ifrach and Johari, 2014; Yin et al., 2014).

We sample the full population (13,195) of popular apps (defined below), and a random sample of 53,457 unpopular apps (from a population of over 1.5 million apps), using rare event sampling. This yielded a sample of 66,652 apps. We collected a second wave of data on these same apps in November 2017 in order to capture time trends and relevant changes. Since we saw almost no category changes, very little change in popularity, and only a 6% change in revenue model⁴ among surviving apps from 2015 to 2017, we focus on the 2015 sample.

We address our research question with a novel and question-driven approach to theory-building theory using large datasets. That is, we combine exploratory data science (Cox, 2017; Dhar, 2013), multiple-case theory-building (Eisenhardt and Graebner, 2007), and machine learning (Athey, 2018; Choudhury et al., 2018; Varian, 2014). Each method both provides its own unique insights and improves the other methods.

Exploratory data analysis (EDA)

We began with “exploratory data analysis” (Behrens, 1997; Cox, 2017; Tukey, 1977). This involves examining data from non-parametric (and often visual) perspectives. The goal is to develop a preliminary understanding of patterns within data (Behrens, 1997). In contrast to “confirmatory data analysis” (i.e., hypothesis-testing), the emphasis is uncovering unexpected relationships and possibly propositions that describe them. EDA begins with broad descriptive categories, and then systematically increases granularity by segmenting the data into smaller sub-

⁴ While some research (e.g., Andries et al., 2013) indicates that business models often benefit from learning and so change, we observe that revenue models are quite stable. One reason may be that there is more to learn with business models given their activity systems. Another may be that users are less tolerant of revenue model changes (e.g., paid to free). Finally, we observe that the most common change was free to freemium with an accompanying increase in product complexity, consistent with our configurational framework. Thanks to a reviewer for noting this.
groups along possibly relevant dimensions. This increasing granularity sharpens patterns that might otherwise remain hidden, and may advance proposition development.

We began by measuring several variables from our App Store data that are likely to be important for our research. We measure the performance of each app with a binary measure, *popularity*. Apple selects up to 240 apps in each category (or subcategory) to feature in a “popular” list. Although the exact selection procedure is not revealed, Apple’s popular ratings are seen by experienced observers to be very closely linked to the number of downloads and commercial success of the product (Bresnahan et al., 2014; Garg and Telang, 2012; Ifrach and Johari, 2014). While no performance measure is perfect (Hallen et al., 2018), ours identifies high-performing product apps in an accurate and relevant way for our research question.

We measure the revenue model of each app, *revenue model*. We assign each app to one of three revenue models that are consistent with extant research, including work using App Store data. Specifically, we assess: 1) *paid* by whether the firm sets an up-front price (between $0.99 and $999.99) that the user must pay before downloading the app, 2) *free* by whether the user can obtain the app without payment (as explained below, this includes the advertising model), and 3) *freemium* by whether the user can obtain the app for free, but can also purchase at least one up-sell product (e.g., on-demand streaming at Spotify). Apple allows up to ten types of such up-sells termed “in-app purchases”. We were able to categorize each app in our entire sample into one of these three revenue models, indicating mutually exclusive and collectively exhaustive categories.

---

5 Popularity as a performance measure is common in AppStore studies because it accurately captures important performance factors like volume (downloads) and commercial success. Our case studies triangulate further with cases where high (e.g., OpenTable, CandyCrush) and low performance (e.g., LumberJack, Medical Visual Books) are quite clear. In contrast, user ratings are not good performance measures (i.e., they measure quality) but do not predict popularity in our data. Finally, given our theory-building aim, we are not testing statistical significance or using precise causal identification that require sharp measures. Rather, a robust measure such as ours is suitable. Our ML models substantially improve revenue model classification among popular products, but not unpopular – consistent with weak performance. A much higher percentage of popular products survive during our study (75% v. 45%), consistent with performance. We appreciate a reviewer’s raising this issue.
We measure the market of each app with the variable, *category*. The App Store uses 22 categories such as Lifestyle, Travel, Sports and Finance with each app product assigned to one category. The developer designates the category, and then Apple verifies this designation and rejects ones with incorrect categories. Since customers browse by category, developers are also motivated to make an accurate selection.

Our next step was successively dividing the data into increasingly granular segments, and creating tables and visuals like pie and frequency charts. First, we segmented and analyzed our data by *revenue model*, and then by performance measure, *popularity*. Increasing our depth, we divided our sample by *category* to observe relative rates of revenue models in each market and at different popularity levels. Next, we sub-divided our data to explore particular categories and revenue models in more depth (e.g., frequency of # of up-sells within freemium apps).

Overall, EDA revealed several surprising patterns. First, popular and unpopular products have very distinct profiles. For example, popular ones are more likely to use freemium models (62%) whereas unpopular ones are more likely to use paid (33%) or free (57%) revenue models (Figure 1). Second, some market categories rely primarily on one revenue model while others use another or a mix, suggesting underlying theoretical distinctions (Figures 2-3). For example, the free revenue model dominates Finance (80%) whereas freemium dominates Games (75%). Third, exploration of specific categories and revenue models revealed other unexpected patterns. For example, by plotting the frequency of # of upsell products within the freemium model (Figure 4), we found an unanticipated bimodal distribution, suggesting that there are actually two distinct freemium models that we term: *fragmented* and *bundled*.

**Multi-case theory building**

We next used EDA insights to guide the theoretical sampling of our multi-case analysis. Given limited theory and conflicting evidence on revenue models (described in Background),
this method is appropriate for our study (Eisenhardt and Graebner, 2007). Multi-case theory building involves a dive into a small number of specific cases (Yin, 2009). The aim is to identify relevant constructs, underlying mechanisms, and theoretical relationships (Eisenhardt et al., 2016). Replication logic is central: Each case is analyzed as a stand-alone entity, and then its insights are systematically compared with each other case in order to iteratively build theory (Yin, 2009). Thus, replication logic is distinct from the pooled logic of econometrics-based research. Unexpectedly, replication logic is instead similar to cross-validation in machine learning (i.e., insights from one case are tested against the others) which similarly lessens overfitting. It is a key reason why multiple cases are more likely to produce better theory – i.e., more robust, parsimonious, accurate, and generalizable – than single cases that tend to over-fit and miss the appropriate theoretical abstraction (Eisenhardt, 1989; Eisenhardt and Graebner, 2007).

We began by selecting cases using theoretical sampling. In contrast with random sampling in econometrics-based research, theoretical sampling is appropriate here because it centers on selection of meaningful cases for building theory, not testing it (Glaser and Strauss, 1967). Specifically, we used patterns identified by EDA. We first selected eight categories that varied on dominant (i.e., most common) revenue model, balancing among categories that tip towards one revenue model and those that do not (e.g., Finance, Travel, Music) and among categories where popular and unpopular choices were more/less the same (Games, Health & Fitness, Food & Drink). Specifically, we chose Games, Photo & Video, Social Networking, Health & Fitness, Music, Travel, Food & Drink, and Finance. Next, we selected popular and unpopular apps with the dominant revenue model in each category. This theoretical sampling of polar types (i.e., extremes) typically clarifies patterns and insights. We also selected a non-conforming popular product (i.e., successful app with a revenue model counter to the dominant category type). This theoretical sampling of counterfactuals helps to eliminate alternative
explanations and improve insights. Within these theoretical sampling frames, we purposefully chose 1) some well-known cases (e.g., OpenTable, CandyCrush, WebMD) in order to more easily obtain data and exploit the clarity of extreme cases, and 2) some randomly selected cases to improve the robustness of our emergent theory.

We repeated this sampling (i.e., 2 polar types, counterfactual, random) for each of the eight categories (i.e., 4 cases each, see Table 2). We also randomly sampled eight popular and unpopular apps across other categories to improve robustness. In sum, we selected 40 cases.

Next, we collected online archival data (e.g., media coverage like TechCrunch, press releases, and videos) for each case, downloaded and examined each product (e.g., content and sample reviews), and conducted primary interviews where possible. Unlike traditional hypothesis-testing research, multi-case theory building does not require identical data across cases (Glaser and Strauss, 1967). Since our research question concerns content (not process) and our sample has few changes (see above), our data for each case are appropriately less rich and more cross-sectional than a typical multi-case study that richly examines process over time.

Consistent with multi-case methods (Eisenhardt and Graebner, 2007), we built each case, and then conducted within-case analysis to address our research question. Next, we moved to cross-case analysis where we further developed tentative constructs, underlying mechanisms, and theoretical relationships from individual cases and compared them across cases (i.e., replication logic). By cycling through emerging theory and data, we strengthened our underlying theory, and clarified constructs and related measures. We brought in prior research to sharpen insights. Once we had a strong correspondence (i.e., theoretical saturation) among data, measures, and theory (Glaser and Strauss, 1967), we finalized the analysis.

Overall, our multi-case theory-building revealed several insights. For example, our analysis identifies constructs (e.g., quality signals and user resources) that are new to the
revenue model research and appear related to effective revenue models. Our analysis also
indicates more precise revenue models (e.g., free, transaction and third-party (including
advertising) within the free model) that add to revenue model research.

**Machine learning (ML)**

We use machine learning to add scale and precision to our multi-case analysis. Broadly,
machine learning (ML) sees empirical analysis as using algorithms to systematically estimate
and compare many alternative models, and then pick the best (i.e., most predictive) (Athey,
2018; Choudhury et al., 2018; Varian, 2014). In contrast, traditional econometrics estimates a
single (or a few) model(s), and then tests coefficient significance. Thus, ML is characterized by
robust models that predict out-of-sample (i.e., generalize) well while traditional econometrics are
characterized by optimized models that fit (or over-fit) a particular data set well (Breiman, 2003;
Choudhury et al., 2018). Strikingly, when ML is used for theory-building, it produces emergent
theory that is similar to that of multi-case methods – i.e., robust, simple, accurate and
generalizable. Further, ML and multi-case theory building are unexpectedly complementary:
Case studies a) indicate relevant constructs that narrow the search space (making ML more
meaningful) (Lettau and Pelger, 2018) and b) provide theoretical mechanisms (complementing
ML’s atheoretical pattern recognition). In complement, ML adds a) the possibility of large-scale
corroboration and elaboration of multi-case theory, and b) precise identification of non-
linearities, equifinal paths, and effect sizes (all challenging to accomplish with case studies).

ML uses two broad techniques to improve prediction and generalizability by limiting
overfitting and complexity (Varian, 2014; Athey, 2018; Choudhury et al., 2018; Jung et al.,
2016). First, **cross-validation** involves systematically (algorithmically) estimating models from
training data and then testing on validation data. This limits over-fitting. A common method is **k**-
fold cross-validation that splits a data set into **k** equal parts. Each **k** part is then successively used
as the validation data and the $k$-1 parts as the training data, resulting in an averaged final estimate. Second, *regularization* involves approaches to “penalize” excess model complexity such as limiting coefficient size in regression-like techniques and limiting the number of leaves on decision trees. Regularization simplifies, and “tunes” the balance between over- and under-fitting the data. Overall, ML emphasizes simple models that predict out-of-sample well, and are robust, accurate, and generalizable. In fact, simple models (e.g., 3 to 6 predictor variables) are surprisingly accurate (i.e., rival the accuracy of complex models) across many data sets (Bell et al., 2008; Jung et al., 2016).

We chose ML over traditional econometrics for several reasons. First, ML makes fewer assumptions about the underlying model that best represents the data (Choudhury et al., 2018; Shmueli, 2010; Varian, 2014). Thus, it more accurately models patterns, and is better able to capture non-linearities and equifinal paths. In contrast, these can be difficult to model with traditional econometrics, given assumptions such as linearity (Breiman, 2003). Second, ML is inherently a theory-building approach to produce simple, accurate and generalizable models that fits our data-driven, theory-building aim (Jung et al., 2016; Puranam et al. 2018; Choudhury et al., 2018). In contrast, traditional econometrics often uses complicated models that fit (or over-fit) specific data in order to test coefficient significance, but may also generalize poorly and so be inconsistent with our theory-building aim. Finally, ML is useful in our study because our data are cross-sectional and our research question is a classification problem, both of which fit the state-of-the-art in the application of ML to theory-building (Athey, 2018; Varian, 2014).

We began by choosing a random sample of 202 popular apps and 202 unpopular ones from our sample of 66,652 apps.\footnote{Prior work indicates that random samples of even 0.1% can} 

---

\footnote{We omit apps with *transaction* revenue models (explained later) since these are sales channels, not meaningful products.}
produce accurate ML results (Varian, 2014). To ensure balanced data for cross-validation, we randomly chose apps within each revenue model. We also collect and set aside a separate random sample of 50 popular and 50 unpopular apps to be the initial test data. Next, we hand-collected measures of variables for each app as suggested by our multi-case analysis. We use App Store and online sources. We added product quality from prior research. Specifically, we added measures for *quality signals, quality, user resources* and *product complexity* and *fragmented* and *bundled* models within the freemium revenue model (Measures in Appendix).

We use three complementary ML methods (Methods in Appendix) (see Choudhury et al., 2018 for a review). First, we use *penalized multinomial logistic regression* to highlight the size and directional effects of predictor variables. This technique differs from traditional multinomial logit: 1) It relies on a *k*-fold cross-validation to determine robust coefficients, 2) it imposes a complexity penalty (regularization) on excessively large coefficients, and 3) it focuses on fewer, but the most important predictors (rather than many predictors and controls) and reports correct classifications (not significance levels) – all consistent with our theory-building aim. By penalizing large coefficients, this technique highlights the most important variables because these will remain large even with penalization.

Second, we compute a *decision tree* to classify the revenue model of each product with a series of true/false decision nodes. Decision trees are effective when important non-linearities, interactions like configurations, and equifinal paths exist (Varian, 2014). Even when decision

---

7 Unbalanced data sets can lead to misleading measures of classification performance. Consider building a model to test for a rare disease (e.g., that appears only 1% of the time). A highly accurate (but poor) test would simply predict “no disease” 100% of the time. This would achieve an accuracy of 99%, but would miss all positive (i.e., “disease”) samples.

8 The test set is randomly selected from the full population (for each popular and unpopular), and is disjoint from our training and validation data (i.e., of 202 popular and 202 unpopular apps). This fully random sampling ensures that the model test accuracies reflect true out-of-sample performance. We omit apps with *transaction* revenue models from the test set since these are not meaningful products (see footnote 6).
trees do not improve prediction (e.g., % correctly classified), they often reveal features of the data that are not apparent in traditional econometrics approaches. The visual output of decision trees is particularly helpful for uncovering and interpreting novel results.

Third, we use a random forest ensemble. This method generates many decision trees (i.e., a random forest) using random sub-samples of data and predictors, and averages these decision trees ($n=100$ in our study) to make a prediction. Random forests provide accurate estimates of the most important predictors because they are based on many uncorrelated decision trees. They are also particularly effective for highly non-linear data such as discontinuities, and for indicating the most important variables in terms of their contribution to prediction accuracy. We use these complementary methods in separate analyses of the popular and less popular apps.

Overall, ML adds several insights including identifying equifinal paths, configurations, and the most important variables to our study. The result of combining EDA, multiple cases and machine learning is the emergent theoretical framework that we present next.

**EMERGENT THEORETICAL FRAMEWORK**

Our research question asks, “*When should particular revenue models be used?”* Broadly, we find that popular v. unpopular have different profiles (Figure 1). Popular apps tend to use freemium (62%) whereas unpopular apps are more likely to use the simpler paid (33%) and free (57%). Consistent with high performance, 75% of popular apps survive during our study whereas only 45% of unpopular apps do. Further, ML often correctly predicts the revenue model for popular apps (about 70% v. naïve estimate of 33%) whereas it weakly predicts the revenue model of less popular apps (about 44%), suggesting inconsistently and perhaps poorly chosen revenue models. We turn now to our framework that describes each revenue model and the configuration of variables related to its performance.

**Free revenue model: Transaction and third-party**
As defined earlier, free is the revenue model in which the user does not pay for the product, and there is no up-sell product. Our EDA indicates that this model is widely used across market categories, particularly by less popular apps (57% v. 35% of popular apps). There is also high variation across categories. For example, about 80% of popular apps in Finance, Travel, and Lifestyle use the free revenue model. In contrast, only about 20-30% of the popular apps in Books, Photo & Video, and Games rely on this model (Figures 1-3). Interestingly, this model appears to be a default choice for less popular apps. Only 10% of unpopular apps using the free model actually possess the characteristics (see below) related to its use by popular apps.

From our multi-case analysis, we observe two types of free revenue models. The first, which we term transaction, occurs when the app provides an efficient and/or convenient sales channel for users to purchase non-app products. Examples include apps for buying from Domino’s Pizza and making a purchase at Target. Here, the app product is a sales channel that provides transaction convenience and/or efficiency. The revenue comes from purchase of the product (e.g., pizza and clothing), not use of the app.

A second and more interesting model is what we term the third-party revenue model. This model relies on actors (other than the user) to pay. Although prior research studies the advertising revenue model (e.g., Chen et al., 2016; Gabszewicz et al., 2005), our cases suggest that this designation is too narrow, and that third-party is the more generalized revenue model. Thus, we subsume the advertising model into third-party.

Our multi-case analysis indicates that the third-party model is likely to be effective when the user provides valuable user resources while using the product. The firm can then sell these resources to a third party that is willing to pay (e.g. advertiser, investor, or bank). User resources contrasts with prior research (e.g., Lin et al., 2012; Casadesus-Masanell and Zhu, 2010; Gabszewicz et al., 2005) that emphasizes advertising rates and ad-aversion. Yet this work misses
the advertiser perspective. In contrast, we observe that user resources are more relevant to advertisers – i.e., advertisers pay high ad rates (and ad-aversion drops) when valuable user resources are available to create better matches between ads and users. Overall, user resources is the important variable associated with effective use of third-party revenue models, including advertising. User resources include financial assets and personal data (e.g., purchase intentions, friendship networks). For example, the user of a travel app reveals that she is researching a trip to New York City in the next month. This is valuable information for an NYC hotel operator that is otherwise unavailable.

The ML models – decision tree, random forest, and penalized multinomial logit – further underscore the importance of user resources. In fact, UserResources is the most important variable for distinguishing the third-party model from the paid and freemium models among popular apps. It is the first (and thus most important) branch of the decision tree and directly predicts the third-party revenue model (Figure 5). User resources has the largest (and positive) coefficient in the penalized multinomial logit, and is the most important “feature” for accurately predicting the revenue model in the random forest ensemble (Table 3).

In contrast with the literature (e.g., Chen et al., 2016; Eckhardt, 2016), neither our multi-case nor ML analyses indicate that product quality is importantly related to the choice of revenue model among popular app products. This holds for all three major revenue models: free, paid and freemium. In contrast, quality is related to the third-party revenue model for unpopular products, suggesting these products miss the importance of user resources and a third-party lens.

A well-known example of a third-party revenue model is WebMD in the Health & Fitness category. The free WebMD app provides physician-reviewed health information, as well as a “symptom checker,” which allows users to search diseases by their associated symptoms. This app tracks specific diseases that concern its consumers. These data are a valuable user resource
for pharmaceutical firms for better targeting their advertising to the right potential consumers. Thus, the information that WebMD gathers from its users is very valuable (and difficult to obtain otherwise) for these firms, making them willing to pay to advertise to these well-targeted users. Conversely, while advertising can annoy (Gabszewicz et al., 2005), advertising that is directly related to health concerns is likely to be tolerated and even appreciated. Estimates place the revenues of this popular app in excess of $700 million (Reuters, 2017).

Another example is OpenTable, a popular product in Food & Drink. OpenTable is a restaurant directory and booking service that enables users to browse and make reservations at a variety of restaurants. An important insight regarding OpenTable is that it gathers valuable information (i.e., user resources) regarding the preferences of potential restaurant goers. By using these data, OpenTable enables restaurants to pay to advertise and makes it attractive to do so (Miller, 2011). Ads can be well-targeted to people who like to eat out (and may prefer specific cuisines and price points), providing favorable matches with advertising restaurants.

Like many firms with this revenue model, OpenTable actually has several third-party revenue models that take advantage of its valuable user resources. A common approach is creating a two-sided marketplace that charges the non-user side a fee. In the case of OpenTable, the firm has two third-party revenue models: restaurant owners can pay for advertising (above) and pay for the booking service in which OpenTable enables users to reserve tables. While a reservation is free for users, OpenTable “sells” its users’ reservations to restaurants for $1 per seat. These third-party revenue models netted OpenTable about $200 million in annual revenue, leading to acquisition by the Priceline Group for $2.6 billion in 2014 (Lachapelle, 2016).

A third well-known example is Robinhood in the Finance category. Robinhood is a stock market investment platform, through which users can buy and sell stocks, free of charge. The data collected on users’ stock preferences are not particularly valuable. However, since users
give Robinhood access to their invested capital, the firm can hold the investors’ funds between trades, earning interest on the principal. In this model, Robinhood “sells” the consumers’ valuable resources (i.e., funds) to a bank, for which it receives revenue (i.e., interest) in return. Robinhood’s valuation stands at $1.3B (Constine, 2017), consistent with the value of the third-party revenue model for this popular app.

There are several reasons why valuable user resources fit with the third-party revenue model. First, since users do not pay, products must ultimately have other actors (i.e., third-party) who are willing to pay. Second, when products offer user resources that are valuable to at least some actors, these actors are likely to pay for them. In the specific situation of advertisers as the third party, user resources (i.e., information) enable better matches with potential consumers (e.g., better targeted ads) and so are likely to increase advertisers’ willingness to pay higher rates. Similarly, better matches are likely to increase user utility (e.g., less ad-aversion) because of greater interest in the advertising content. Finally, these reasons are also consistent with the observation that the third-party revenue model is common in Lifestyle, Food & Drink and Travel (where users typically reveal personal data including purchase preferences and intentions), but rare in Games and Photo & Video (where users often provide few, if any, valuable resources such as invested capital and personal information for well-targeted advertising).

While user resources are importantly associated with third-party revenue models among popular products, they are much less so in unpopular ones (Table 4). Our data indicate that only 10% of unpopular apps with third-party revenue models provide valuable user resources. User resources is not even a decision node on the decision tree for unpopular apps (in striking contrast to popular apps), suggesting that many of these less popular products are using an ill-fitting revenue model (Figure 6). An example is Coub, a less popular product in Social Networking. Its product – a broadcast service for short videos (ten seconds or less) – entertains, but gathers little
user information because its videos are short and difficult to interpret. Similarly, its connections (“subscribers”) are not based on close ties that reveal personal information. Coub has been unable to gain traction with advertisers, contrasting with products like Facebook that gather and sell valuable user information to various third-parties.

Finally, our multi-case analysis reveals the core challenge of the third-party revenue model – i.e., recognizing valuable user resources and finding a third-party to pay for them. For example, LinkedIn in the Social Networking category quickly added users who were interested in keeping up with their professional friends. While VC investors were initially pleased, they ultimately demanded revenue (Piskoski, 2007). Yet LinkedIn struggled to figure out what was valuable about their many users and for whom. By carefully segmenting its users, LinkedIn stumbled into the realization that the third-party willing to pay was employers seeking new hires (a very small percentage of LinkedIn users), and the valuable user resource was information about job skills that virtually all users reveal on their profiles (Piskorski, 2007).

**Paid revenue model**

As defined earlier, the paid revenue model is one in which the user pays an upfront price. Our EDA indicates that the paid revenue model is used less frequently than the others, especially among popular products. For example, only 3% of popular ones use this model. In contrast, a much higher percentage of less popular apps (33%) uses the paid revenue model. Finally, usage among popular apps is highest in Weather, Health & Fitness, and Books, but even here usage is typically only about 10%.

Our multi-case analysis indicates that the paid revenue model is effective when the product has substantial **quality signals**, especially early on. This analysis also indicates several ways in which firms gain quality signals. First, firms can signal product quality through media coverage, including of awards. This coverage can build a positive view of the product among
users prior to purchase. An example in Games is Monument Valley (a counterfactual case – i.e., paid revenue model in a category dominated by freemium). Monument Valley is a puzzle adventure game first released in 2014. The game boasts unique Escher-inspired graphics that distinguish it from other games (Lomas, 2013). Indeed, the developer spent a significant amount – over $800,000 – to develop Monument Valley (Etherington, 2015). With the release of a trailer and substantial media attention, the game was highly anticipated before its launch (Lomas, 2014). Given its distinctive look and play, Monument Valley won several prestigious awards upon release, including both Apple’s Worldwide Developers Conference Design and Game of the Year awards (Loyola, 2014; Spencer, 2014). These triggered additional and significant early media attention. Its developer, Ustwo, successfully chose a paid revenue model.

Firms can also signal product quality when the firm itself is a popular developer – i.e., one that has previously developed popular products. For example, the developer Fitness 22, founded in 2011, focuses on Health & Fitness products. The firm explicitly eschews expenditure on marketing. Rather, the company concentrates on building simple, reliable apps, and has a track record of popular products (Bort, 2015) at a range of prices. In fact, five of Fitness 22’s apps have reached the popular rankings. Following its prior successes, Fitness 22 released its next product, 5K Runner, with a paid revenue model and found similar success (Bort, 2015).

Finally, firms can signal quality when related products have an established presence in other venues. For example, this often occurs for video games that are ported from other platforms such as PCs or consoles, and for related publications that are well-known in the print or online media. For example, National Geographic released its popular World Atlas app in the Reference category, after decades of publishing its well-regarded maps and magazines, both online and in print. National Geographic’s long-established presence in other venues signaled quality, consistent with its choice of the paid revenue model for World Atlas.
In contrast, when products use the paid model but are unable to signal quality, they are more likely to be unpopular. For example, Health & Fitness product, *How to Surf Like a Pro*, lacked quality signals – i.e., it received no media attention, was not developed by a firm with past popular apps, and did not have related products established in other venues. Similarly, less popular products such as *Yes!! I Know Plumbin* (Reference), *Ballet Dancing* (Music), and *Stem Cell Therapy* (Medical) also used the paid revenue model without quality signals.

The ML analyses underscores the importance of quality signals for the paid revenue model, and further reveals that *Media* is the quality signal most consistently related to the paid revenue model (Table 3). For example, the penalized multinomial logit indicates that media has the highest coefficient among quality signals in distinguishing the paid revenue model. The random forest finds all three quality signals – *Media, OtherVenue*, and *PopularDeveloper* – to be important. The decision tree shows that, while valuable user resources is the most important feature distinguishing the third-party revenue model v. other models (prior section), quality signals (i.e., popular developer and media) are the next branches – i.e., most important features distinguishing freemium v. paid revenue models (Figure 5). Thus, quality signals – *Media, OtherVenue*, and *PopularDeveloper* – are associated with the paid revenue model, with media as the most consistently the most predictive.

One reason why quality signals are associated with a paid revenue model among popular products is that firms realize that these signals are particularly salient to users. When quality signals are present, users may be more willing to purchase the product without trying it first. In contrast and contrary to prior research (e.g., Chen et al., 2016; Eckhardt, 2016), quality has surprisingly little relationship to the choice of any revenue model among popular products. For example, it is the least important feature in the random forest and has an almost zero coefficient in the penalized multinomial logit. It is absent in the decision tree, suggesting that it is not highly
predictive of any revenue model. Users appear more attuned to quality signals (i.e., media, past developer success, and well-known related products) as more observable, salient, and relevant.

Finally, unpopular products are strikingly more likely to use the paid revenue model than popular ones (33% v. 3%). The decision tree for unpopular products also indicates that low quality and paid revenue models are related, suggesting developers of these products are overly optimistic about them (Figure 6). In the penalized multinomial logit, Media, followed by OtherVenue, is predictive of the paid revenue model for unpopular apps (Table 4). This is supported by the decision tree in which Media is the second node following Quality. However, while the paid model is predicted by the media variable, only a small number of products are classified in this node of the decision tree. This suggests that few unpopular paid products receive media attention.

A special case of the paid revenue model is subscription in which payment recurs.⁹ For firms, subscription is attractive because users repeatedly pay. In contrast with one-time payment (including per-use pricing), subscription is associated with frequent change in the product that users appreciate. Nonetheless, the logic for the subscription model is the same as for the broader paid revenue model – i.e., users are repeatedly willing to pay because of continuing quality signals. An example is Netflix, a popular product in Entertainment. Consistent with its monthly subscription fee, it reliably signals continuing quality with its track record of engaging TV and film content that it consistently refreshes each month. Netflix also signals continuing quality with its personalized suggestions that become increasingly accurate.

---
⁹ Given space limits and infrequency of paid revenue models among popular products, we do not delve into paid models such as flat fee v. usage pricing. Instead, we focus on revenue models (e.g., freemium) that are better represented in our data. That said, we expect that these variations of paid are also likely related to quality signals as per our framework. This could be explored in future research.
In sum, our analyses indicate that gaining **quality signals** (e.g., **media**) is the core challenge of the paid revenue model while **quality** per se is less relevant. Further, largely independent of market category, the paid model is effective for only a few select products.

**Freemium revenue model: Bundled and fragmented**

As described earlier, **freemium** is a revenue model in which users can use the product without charge, but also pay for up-sells that unlock digital content within the product itself. The freemium model is the most common revenue model among popular products (62%). By contrast, the freemium model is used much less by unpopular ones (11%), suggesting that firms offering less popular apps avoid complex revenue models (i.e., freemium) in favor of simpler revenue models, **paid** and **free**. Photo & Video, Magazines, and Games are the most common market categories for popular products using the freemium model. A core challenge is deciding what is free and what is not, and broadly how to modularize the product to make this distinction.

From our EDA, we observed two types of freemium revenue models. In the first, which we term **bundled**, the product is free up-front with one or perhaps a few up-sells. The bundled model is used in popular products that are **complex** – i.e., many interrelated and reinforcing features (i.e., super-modularity) that users frequently combine when using the product. In this situation, it is difficult to separate the premium features of the product, and to sell them separately without losing substantial value. Thus, up-sells are combined into one or maybe two mutually reinforcing bundles that are major upgrades with lasting value. Users use these bundles repeatedly and together, rather than as single-use consumables.

A well-known popular example of bundled freemium is **Spotify**, a streaming service in the Music category. Spotify users can listen to shuffled music free of charge, but they can also purchase an up-sell (i.e., a premium product) with several features including on-demand streaming (i.e., user can play a specific song) and offline mode (i.e., users can download songs to
their devices). These features are highly interrelated and more useful together than separated. For example, downloading a single song is useless without the ability to play that individual track. These features also improve the overall product value over the long run because the Spotify service learns which songs users play, and so provides increasingly better new music recommendations. Spotify continues to add subscribers – in 2016 the firm surpassed 60 million subscribers and $3B in revenues (Plaugic, 2017; Statista, 2018a).

A randomly selected example of the bundled freemium revenue model is *iTranslate*, a popular product in Productivity. *iTranslate* lets users translate phrases across 90 languages, and includes voice output and bookmarking capabilities. Users can obtain a “slim” product for free, but can also purchase an up-sell. This premium product includes voice input, offline translation, and verb conjugation features that are mutually reinforcing. For example, the value of having voice input increases if it can be used offline while the value of offline translation increases if there are easier ways to use it – such as through voice input. Bundled freemium has been successful for *iTranslate* – it was named *Best of the Year* and an *App Store Essential* by Apple, and is a top grossing Productivity app (*iTranslate*, 2018).

In contrast, less popular apps tend to offer many up-sell products even when their features are highly interrelated (i.e., complex). Yet, fragmenting inter-related up-sell products places too many features behind separate paywalls. This makes it difficult for users to understand which features they might prefer. It is also frustrating for users to face extra steps and payments. As a result, they are often less likely to purchase the up-sell product. For example, *Medical Visual Books* in the Medical category provides human body anatomy graphics. It offers many up-sells, including *3D Body Anatomy*, *2D Skeleton System*, and *Lymphatic System* visuals. Since most of the app’s features are separated into many small up-sells, it is difficult for users to
assemble and experience the value of these interrelated human systems in the complete product. Few consumers choose to upgrade, and this unpopular product has languished.

In sum, bundled freemium is an effective revenue model when the core and up-sell products consist of many tightly linked and reinforcing features (i.e., complex). Here, breaking up the up-sell product into many pieces is difficult (e.g., where to draw the boundaries between chunks?), and reduces the overall product value. Implicit in making bundled freemium work is articulating and demonstrating the benefits of the bundle (i.e., up-sell). A core challenge is striking a balance between the free product and paid premium upgrade. If the free version is too simple with too few features, users will not experience enough benefits to pay for the up-sell product. Conversely, if the free version is too inclusive, they will not need to upgrade.

A second freemium revenue model is what we term fragmented freemium. Here, the firm offers the product free up-front, but up-sells many “fragmented” products. That is, many additional features, content and other products are each sold separately. Indeed, as we unexpectedly discovered in our EDA, there is a sharp bifurcation of the number of up-sells (i.e., bimodal distribution), indicative of two distinct models (Figure 4). Our multi-case analysis indicates that firms choose fragmented freemium when (while the core product is often complex) they can easily added multiple and diverse product features that they can be offered individually. We also noted that these added features are often consumables – i.e., in-app purchases that are depleted after use. This model is effective, in part, because it allows users to self-select different up-sells based on their interest and willingness to pay.

A well-known example is Candy Crush Saga in Games. Candy Crush Saga is a puzzle arcade game in which players move candy pieces on their screens to create 3-of-a-kind matches, earn points, and beat levels. When players lose, the game times-out, and they must wait 24 hours before playing again. Candy Crush offers many in-app purchases such as extra lives, additional
moves, and power-ups with which users can avoid the timeout period. The high variety of up-sells means that consumers can choose the add-ons that they prefer or need most. Since these add-ons are modular and prices are low, users can easily and quickly buy these offerings. The core Candy Crush game is highly complex (i.e., many interrelated features), but its extra features (e.g., extra lives, moves, and power-ups) are modular consumables, not closely interrelated. Consequently, fragmented freemium is an appropriate revenue model. Using this revenue model, *Candy Crush Saga* generated over a billion dollars in revenues in 2013 (Statista, 2018b).

Another example is *Udemy* in Education. It offers online courses on a wide range of topics, such as writing and coding. Using the free version, consumers can browse courses, read reviews, and preview lectures. Yet Udemy up-sells individual courses that are highly modular – i.e., from $9.99 for short courses, up to $199.99 or more for high-quality, certified courses. The natural segmentation of courses into individual up-sells and their consumable character makes fragmented freemium an effective choice. For example, in 2016, Udemy reported over 10 million students taking courses across 190 countries (Udemy, 2016).

A core challenge with the fragmented freemium model is developing the right up-sells. A well-known example is *Snapchat* in Photo & Video. Snapchat has struggled to find up-sells for its freemium revenue model. In 2016, the company introduced animated photo filters for $0.99, and photo replays for $4.99. Yet even with these low prices, these consumable up-sells were unsuccessful and quickly dropped (Valinsky, 2016). Recently, Snapchat’s location-based consumable geofilters has seen some traction, but the company has yet to succeed.

In sum, the fragmented freemium model works well when, although the free product is often complex, the premium (i.e., up-sell) products are modular such that added features can be offered. A core challenge is finding many multiple up-sells that attract users and lock them in. Another is managing the complexity of these up-sells.
Finally, our ML analysis adds insights into freemium revenue models. The decision tree for popular products identifies two equifinal paths to the freemium revenue model (Figure 5). As per above, UserResources separate third-party revenue models from paid and freemium. With low user resources, there are then two paths to freemium. In the first path, successful firms (PopularDeveloper > 0.5) with significant media attention (average Media ≈ 0.8) have a choice between paid and freemium. These successful developers have the quality signals to use the paid model and the experience to handle the complexity of freemium. In the second path, firms with products that lack quality signals (needed for paid) and user resources (needed for free) are forced into freemium – i.e., freemium is their choice of last resort. The penalized multinomial logit results are consistent (Table 3). UserResources, OtherVenue, and Media negatively predict freemium while PopularDeveloper positively predicts it.

We further clarify bundled v. fragmented freemium with added ML analyses (Table 5) where we examine bundled v. fragmented given a freemium revenue model. The penalized multinomial logit highlights that popular bundled freemium products are more likely to have Interrelated premium (up-sell) features while fragmented freemium products are likely to have many Consumable up-sell features. In fact, the Consumable feature is the most predictive – i.e., largest coefficient in the penalized multinomial logit, and first in the random forest. It is the first (and only) decision node in the decision tree, further highlighting its importance (Figure 7).

**DISCUSSION**

Intrigued by the puzzle of contrasting revenue models for Spotify v. Pandora, we asked *when should particular revenue models be used?* Our core insight is an emergent theoretical framework that indicates the optimal choice of revenue model (Table 1). Prior research identifies the importance of product quality (Chen et al., 2016; Eckhardt, 2016; Prasad et al., 2003), ad-aversion (Gabszewicz et al., 2005; Lin et al., 2012; Riggins, 2002), and advertising rates
(Casadesus-Masanell and Zhu, 2010; Lin et al., 2012). Yet, this research, while useful, is often incomplete (e.g., misses the advertiser lens), too simple (e.g., under-theorizes freemium models) and conflicting (e.g., inconsistent results for product quality). More deeply, it misses possible equifinal paths and configurations of variables. As such, it leaves open an increasingly important strategic puzzle (Massa et al., 2017) – when should revenue models be used.

Using a novel theory-building method that blends machine learning, multi-case theory building and exploratory data analysis, we contribute a new theoretical framework for configurations of effective revenue models (Table 1). Relying on large-scale empirical analysis and small-scale case studies, we study a wide swath of products in 22 market categories in the App Store. Our framework proposes new theoretical relationships that tie quality signals, user resources, and product complexity to the choice of optimal revenue model. Further, we expand the free revenue model, and contribute insights into revenue models like paid that go beyond prior research. We also unpack the under-theorized freemium model by adding both equifinal paths and two distinct models – i.e. bundled and fragmented. Overall, we use question-driven research to contribute a more accurate and theoretical understanding of revenue models.

**Emergent framework of revenue model configurations**

Our core contribution is an emergent framework that describes several configurations of effective revenue models. There are three broad arguments. First, firms should use the **free revenue model** when there is at least one actor (not the user) willing to pay. We further divide the free model into two types: transaction and third-party. The **transaction** revenue model is effective when the app is simply a sales channel that makes purchasing a non-app product more

---

10 As described earlier, we use known constructs in new relationships with new theoretical arguments and dependent variables, consistent with a research contribution, especially in question-driven research.
efficient or convenient. For example, retailers such as Target use a transaction revenue model to facilitate the online purchase of their non-app products like clothing.

The more interesting third-party revenue model is effective when the focal product’s users provide resources that are valuable to another actor. **User resources** (e.g., personal data, revealed preferences, financial assets) provide value for which a third party (e.g., advertiser, employer, investor, seller on a two-sided marketplace) is willing to pay. The third-party revenue model extends prior research in two ways. First, it recognizes that advertisers are simply one type of third-party actor who may be willing to pay for user resources. Second, it recognizes the relevance of user resources for when to use this model. So while prior work argues that ad-aversion (e.g., Gabszewicz et al., 2005; Lin et al., 2012), product quality (e.g., Chen et al., 2016; Eckhardt, 2016), and advertising rate (e.g., Casadesus-Masanell and Zhu, 2010; Lin et al., 2012) should affect the advertising revenue model, we clarify that the more fundamental variable is user resources. We also contribute the core insight that user resources are particularly germane: they distinguish when to use the free revenue model v. paid and freemium.

In addition, our data indicate that the third-party revenue model is the most common choice. Yet, this frequency is driven by unpopular products that often lack the requisite user resources. Indeed, several of our case interviews suggest that some developers of unpopular apps simply assume that others like advertisers will pay. They under-estimate the core challenge of the third party model: figuring out which user resources (if any) are valuable and for whom.

Second, when firms have quality signals for a focal product that lacks valuable user resources, they should use the **paid revenue model**. This model enables the firm to capture the entire revenue stream rather than unnecessarily lose some revenue to a free product (i.e., freemium model). Further, our data indicate three paths to obtaining the necessary **quality signals**. Products can gain quality signals by capturing media attention such as by winning
awards. They can also gain quality signals when their developer has past popular products and by having related products established in other venues like print. Overall, while prior research often emphasizes quality per se (Casadesus-Masanell and Zhu, 2010; Chen et al., 2016; Eckhardt, 2016; Prasad et al., 2003), quality signals are more important because they are likely more observable, salient, and compelling. Indeed, quality (measured by user ratings) contributes little to the choice of optimal revenue model.

In addition, our data indicate that firms with less popular products are likely to use the paid revenue model, even when they lack the requisite quality signals. Developers of these unpopular products seem to miss the importance of quality signals, and be overly optimistic about their products. As such, they under-estimate the core challenge of the paid revenue model: providing enough quality signals to persuade potential users to buy the product.

Third, when firms lack valuable user resources and have few quality signals, they should choose the freemium revenue model. This model enables the product to attract users with a free offering, and then earn revenue with up-sells. Thus, it avoids the problems of over-estimating quality signals (paid model), and missing valuable user resources (if any) (free model). In addition, our data indicate two equifinal paths to freemium. One is for popular developers with past popular products. They could also choose the paid model, but may prefer freemium for reasons like access to a potentially bigger audience. These developers are also likely to be experienced enough to incorporate the requisite complexity and modularity into the product. The other path is for developers who have no choice – i.e, freemium is the last resort for products that lack quality signals and user resources. Overall, our framework offers a nuanced view of the freemium model (i.e., equifinal paths and multiple freemium models) that may clarify why prior research (e.g., Arora et al., 2017; Casadesus-Masanell and Zhu, 2010; Prasad et al., 2003; Rietveld, 2018) has inconsistent results.
Our data also indicate that freemium is the most common revenue model among popular products. In contrast, it is the least common among unpopular ones, suggesting that perhaps less skilled (or overly optimistic) developers prefer to avoid the complexity of the freemium model in favor of simpler models. In so doing, they avoid the core challenge of freemium models: determining where to segment a product between free and paid.

In addition, while prior research usually does not distinguish among freemium models (Casadesus-Masanell and Zhu, 2010; Prasad et al., 2003; Rietveld, 2018), we contribute by adding two types of freemium models with different product complexity. Bundled freemium (i.e., initial product is free, and one or a few up-sell products can be purchased) works well when the product has many interrelated features (i.e., complex) that the consumer uses together and benefits from as a whole (i.e., super-modularity). One or a few highly-interrelated up-sells provide major long-term value. In contrast, fragmented freemium (i.e., initial product is free and many up-sell products can be purchased) works well when the product itself functions, but is enhanced by modular offerings that are depleted after use (i.e., complex core but highly modular periphery). Many and diverse (often consumable) up-sell products work well in attracting and then locking-in users who self-select up-sells based on their interests and willingness to pay.

In summary, our primary contribution is an emergent framework of configurations of effective revenue models (Table 1). Using a novel theory-building approach for question-driven research, we contribute by identifying important variables associated with particular revenue models: user resources, quality signals, and product complexity. We also contribute the core insight that user resources create the primary distinction between third-party revenue models v. paid and freemium. Further, we contribute to the freemium model by identifying two equifinal paths and two distinct models with differing types of product complexity. Overall, while prior research emphasizes linear models and variables like product quality (e.g., Chen et al., 2016;
Eckhardt, 2016), advertising rates (e.g., Casadesus-Masanell and Zhu, 2010; Lin et al., 2012), and ad-aversion (e.g., Gabszewicz et al., 2005; Lin et al., 2012), we contribute a non-linear, novel, and accurate view of effective revenue model configurations.

**Toward a novel theory-building method**

We also make several methods contributions. Our primary one is a novel **theory-building approach** for question-driven research. It uniquely combines multi-case theory-building with machine learning and exploratory data analysis. **Multi-case theory building** has a long tradition within strategy and organization theory (Glaser and Strauss, 2007; Eisenhardt, 1989; Eisenhardt and Graebner, 2007; Yin, 2009). Yet since **exploratory data analysis** can reveal preliminary patterns, it can improve multi-case theory building by providing helpful guidance for theoretical sampling of cases. **Machine learning** is an emerging method within strategy (Choudhury et al., 2018; Puranam et al., 2018). Yet since multi-case theory building produces empirically grounded theoretical constructs, it can improve machine learning with useful initial seeding. This has two advantages. First, this theoretical guidance narrows the search space and so improves model performance (Lettau and Pelger, 2018). Moreover, it likely does so without sacrificing much accuracy. As prior research across many data sets shows (Bell et al., 2008; Jung et al., 2016), simple models with a few very important variables (such as ours) usually rival the accuracy of much more complicated models. Second, this guidance produces interpretable models based on meaningful variables (in contrast with many ML models). In addition, multiple cases provide theoretical mechanisms that compensate for ML’s being inherently atheoretical. In sum, multiple cases provide simple theory – i.e., empirically grounded constructs and theoretical mechanisms (Eisenhardt and Graebner, 2007). ML complements with empirical scale, effect sizes, and precise capture of equifinal paths, non-linearities and configurations (Choudhury et al., 2018; Varian,
2014). Together, these methods provide a novel yet powerful approach for building accurate, parsimonious, and generalizable theory.

A second methods contribution is insight into using machine learning for theory building. Following Varian (2014), we note that ML is not restricted to “big data”, but rather can also be highly effective with much less data. In addition, we observe that (given the state-of-the-art) ML works well for classification problems (such as ours) where time effects (if any) can be modeled in cross-section (Athey, 2018; Varian, 2014). Most importantly, we indicate a useful suite of complementary machine learning approaches. *Decision trees* are especially insightful for revealing key non-linearities like configurations, branches, and equifinality (James et al., 2013; Varian, 2014). They often reveal otherwise hidden features. *Random forests* are especially insightful when there is high non-linearity, and for indicating the most important variables (Varian, 2014). *Penalized multinomial logit* is insightful for revealing the size effects and direction (James et al., 2013). While other approaches exist (see Choudury et al., 2018 for a review), our approaches are interpretable, straightforward, and complementary.

A third methods contribution is identifying the unexpected similarities of multi-case theory building and machine learning (Table 6). They are complementary, yet similar, methods. Both focus on building accurate, simple, and generalizable theory while guarding against overfitting and complexity. Specifically, both methods use analytic techniques to reduce overfitting – i.e., *replication logic* for multiple cases and *cross-validation* for ML. These techniques continually revisit different cuts of the data to generate theory, and then test that theory on the remaining data. They reduce “noise” and help to determine the “best” (i.e., most predictive) model. In addition, both methods cut complexity – i.e., replication logic limits complexity by adjusting the *construct abstraction level* to produce simple and generalizable, yet accurate, theory. Similarly, ML limits complexity with *regularization* techniques like reducing
large coefficients and pruning leaves on a decision tree. In contrast, their counterparts – i.e., single cases and traditional econometrics methods - tend to over-fit data. Single cases offer rich but idiosyncratic descriptions that often produce overly complex theory while traditional econometrics enables significance tests of coefficients but often predicts poorly. Thus, we highlight the surprising similarity between multiple-cases and machine learning.

**Boundary conditions, alternative explanations, and future research**

Like all research, our research occurs in a specific setting that may create boundary conditions. A potential one is digital goods. As noted above, the App Store is a useful setting: Digital goods are a large, growing segment of the global economy, and the App Store covers many products and market categories, and offers clear, accurate measures. Yet as the infrequent use of the paid revenue model by popular products implies, digital and physical goods (where the paid model is common) are different. Nonetheless, our framework may still apply because it describes configurations of variables (e.g., quality signals, user resources, and revenue models). So while the frequency of a specific configuration may vary across types of goods, the framework itself may still hold. For example, quality signals may be stronger and clearer for physical goods, thus increasing the frequency of the paid model and its related configuration of variables, as predicted by our framework. In contrast, digital goods may have more challenges in conveying quality signals, making the paid revenue model infrequently an effective choice. Nonetheless, this is an important area for future research.

Another potential boundary condition is product simplicity that may limit the range of observable revenue models. For example, industries like finance may have more complex

---

11 We appreciate reviewers’ raising possible boundary conditions and alternative explanations such as the limits of App Store products, strategic contexts like subsidizing within product portfolios and venture-backed firms, revenue model complexity such as in finance, contingencies such as geography (e.g., wealth differences), time period imprints and industry structure that might shape configurations and revenue models. Many of these offer fruitful avenues for future research.
products and revenue models. Yet, the residential solar industry provides a possible counter-example. When the industry was disrupted by tax law changes, firms like SunRun developed a financially innovative revenue model – i.e., give free electricity to homeowners, but sell their resources (i.e., rooftops where solar panels could be mounted) to third-party investors who could benefit from clean energy tax credits (Hannah and Eisenhardt, 2018). While the details of this revenue model are novel (and even complex), it is still at its core a third-party revenue model. Broadly, it may be that the revenue models and theoretical logic of simple digital products may actually fit complex products like residential solar and/or are the building blocks of more complicated revenue models such as dynamic pricing (e.g., surge pricing in ride-sharing). This is an intriguing avenue for future research.

Another potential boundary condition is strategic context that may affect the optimal revenue model. One such context is firms with product portfolios such that it is effective to give away (or price very low) some products to boost the revenue of other products or the portfolio as a whole. An illustration is the well-known “razor and blades” revenue model. Yet this is at its core a fragmented freemium model – i.e., the “razor” has a very low price and revenue comes mostly from selling many “blades”. Another strategic context is ventures where VCs may indicate that the best revenue model is none – i.e., give away the product (e.g., to buildup users). Twitter is one of many examples. As suggested, however, by our Snapchat and LinkedIn examples above, this is unsustainable for an independent firm. Indeed, this is more accurately a temporary stage in revenue model design as the focal firm tries to find an effective model. LinkedIn, for example, did succeed in creating a paid revenue model while Snapchat is still trying to design an effective freemium model with up-sells (e.g., photo replay) that have not yet taken off. Overall, it may be that our framework fits a broader set of strategic contexts than our study examines. This is an intriguing question for future research.
Alternative explanations are also important. For example, time periods, geographies, and industries may imprint specific revenue models or make some revenue models more likely. To illustrate, freemium revenue models may be more likely than paid models in recent years, products in developing countries (where users are less wealthy) may rely more on third-party revenue models, or industry regulation may limit revenue models for healthcare. Yet while these exogenous and other factors may affect the likelihood of a specific configuration of revenue model and variables (as we saw in market categories), our framework may nonetheless hold. That is, these factors may affect the frequency of a particular revenue model configuration, but not revenue model directly (see physical goods example above). That said, these and other alternative explanations could be explored in the future.

CONCLUSION

We began by observing the importance of Spotify’s revenue model as a novel source of innovation. We also noted that value creation increasingly involves users and complementors, not just the focal firm, making revenue models more strategically significant (Massa et al., 2017). Given limited theory and conflicting evidence, we pursued question-driven research with a novel theory-building approach. Our primary contributions are twofold: theoretical framework of configurations of effective revenue models and a novel theory-building method combining machine learning, multi-case analysis, and exploratory data analysis. The next steps are to test our framework in more product and strategic contexts.

REFERENCES


Liu, CZ, Au, YA, Choi, HS. 2012. An empirical study of the freemium strategy for mobile apps- Evidence from the google play market.


TABLE 1: EMERGENT THEORETICAL FRAMEWORK: REVENUE MODEL CONFIGURATIONS

<table>
<thead>
<tr>
<th>Revenue model</th>
<th>Variables</th>
<th>Core challenge(s)</th>
<th>Prior Research</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>User resources (High)</td>
<td>Recognizing valuable resources Finding a third-party willing to pay for them</td>
<td>Advertising</td>
<td>Chen et al., 2016</td>
</tr>
<tr>
<td>Third-party</td>
<td>Quality signals (High) Media coverage Prior successful products Established products in other venues</td>
<td>Providing enough quality signals to persuade potential users to purchase</td>
<td>Paid Product quality (Conflicting) Advertising rate (Low)</td>
<td>Duan et al., 2008 Eckhardt, 2016 Casadesus-Masanell and Zhu, 2010 Gabszewicz et al., 2005 Lin et al., 2012 Prasad et al., 2003</td>
</tr>
<tr>
<td>Transaction</td>
<td>Efficiency/Convenience (High)</td>
<td>Fulfillment logistics</td>
<td>Paid Advertising rate (High) Ad-aversion (Low) Product quality (Low)</td>
<td></td>
</tr>
<tr>
<td>Paid</td>
<td>User resources (Low)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freemium</td>
<td>User resources (Low) Equifinal paths (#1 Product with prior popular products or media, #2 Product w/o quality signals)</td>
<td>Optimally separating the free (or almost free) and paid products</td>
<td>Freemium Under-theorized w/mixed results</td>
<td>Arora et al., 2017 Casadesus-Masanell and Zhu, 2010 Prasad et al., 2003 Rietveld, 2017</td>
</tr>
<tr>
<td>Bundled</td>
<td>Complex (i.e., Interrelatedness) (High)</td>
<td>Balancing between attractive free product and one or two valuable paid products</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fragmented</td>
<td>Moderately Complex (i.e., complex core w/ modular periphery)</td>
<td>Creating attractive free product Finding appealing, modular and diverse paid products Managing scale and complexity of many diverse up-sells</td>
<td></td>
<td></td>
</tr>
<tr>
<td>App</td>
<td>Category</td>
<td>Sampling</td>
<td>Revenue model</td>
<td>Sources</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------</td>
<td>---------------</td>
<td>---------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Candy Crush Saga</td>
<td>Games</td>
<td>Well-known popular</td>
<td>Fragmented freemium</td>
<td>3 online interviews 9 online articles Examination</td>
</tr>
<tr>
<td>Monument Valley</td>
<td>Games</td>
<td>Counterfactual popular</td>
<td>Paid</td>
<td>5 online interviews 6 online articles Examination</td>
</tr>
<tr>
<td>LEGO Star Wars: TCS</td>
<td>Games</td>
<td>Random popular</td>
<td>Fragmented freemium</td>
<td>6 online articles Examination</td>
</tr>
<tr>
<td>LumberJack Cut The Beanstalk</td>
<td>Games</td>
<td>Random unpopular</td>
<td>Fragmented freemium</td>
<td>2 online articles Examination</td>
</tr>
<tr>
<td>Robinhood</td>
<td>Finance</td>
<td>Well-known popular</td>
<td>Third-party free</td>
<td>6 online interviews 6 online articles Examination</td>
</tr>
<tr>
<td>iXpenseIt</td>
<td>Finance</td>
<td>Counterfactual popular</td>
<td>Paid</td>
<td>5 online articles Examination</td>
</tr>
<tr>
<td>Mortgage by Zillow</td>
<td>Finance</td>
<td>Random popular</td>
<td>Third-party free</td>
<td>3 online interviews 5 online articles Examination</td>
</tr>
<tr>
<td>FREE Gold + Silver Watch</td>
<td>Finance</td>
<td>Random unpopular</td>
<td>Third-party free</td>
<td>1 online article Examination</td>
</tr>
</tbody>
</table>
TABLE 3: POPULAR PRODUCTS: PENALIZED MULTINOMIAL LOGIT AND RANDOM FOREST ANALYSES.

<table>
<thead>
<tr>
<th>Revenue model</th>
<th>Media</th>
<th>Other Venue</th>
<th>User Resources</th>
<th>Quality</th>
<th>Popular Developer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free (3rd party)</td>
<td>0.268</td>
<td>0.600</td>
<td>1.977</td>
<td>-0.050</td>
<td>0.562</td>
</tr>
<tr>
<td>Freemium</td>
<td>-0.568</td>
<td>-0.225</td>
<td>-0.884</td>
<td>0.055</td>
<td>0.122</td>
</tr>
<tr>
<td>Paid</td>
<td>0.301</td>
<td>-0.375</td>
<td>-1.093</td>
<td>-0.005</td>
<td>-0.684</td>
</tr>
</tbody>
</table>

Test Accuracy (i.e., % Correctly Classified) = 68%

Penalized multinomial logistic regression: Coefficients (log-odds) fit on popular apps \((N=150)\).
Optimal hyperparameter (inverse of regularization strength), \(C = 1\).
Training accuracy = 65%. Cross-validation accuracy = 64% (standard deviation of 7%).

<table>
<thead>
<tr>
<th>Feature</th>
<th>Importance (mean impurity decrease)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserResources</td>
<td>0.666</td>
</tr>
<tr>
<td>OtherVenue</td>
<td>0.103</td>
</tr>
<tr>
<td>PopularDeveloper</td>
<td>0.100</td>
</tr>
<tr>
<td>Media</td>
<td>0.073</td>
</tr>
<tr>
<td>Quality</td>
<td>0.057</td>
</tr>
</tbody>
</table>

Test Accuracy (i.e., % Correctly Classified) = 68%

Random Forest: Predictor importance \((n_{trees}=100)\) analysis on popular apps \((N_{apps}=150)\).
Optimal hyperparameters: minimum samples at leaf = 10, maximum depth = 4.
Training accuracy = 65%. Cross-validation accuracy = 64% (standard deviation of 9%).
TABLE 4: UNPOPULAR PRODUCTS: PENALIZED MULTINOMIAL LOGIT AND RANDOM FOREST ANALYSES.

<table>
<thead>
<tr>
<th>Revenue model</th>
<th>Media</th>
<th>Other Venue</th>
<th>User Resources</th>
<th>Quality</th>
<th>Popular Developer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free (3rd party)</td>
<td>0.629</td>
<td>0.167</td>
<td>0.365</td>
<td>0.685</td>
<td>0.162</td>
</tr>
<tr>
<td>Freemium</td>
<td>0.407</td>
<td>1.468</td>
<td>0.869</td>
<td>1.356</td>
<td>4.648</td>
</tr>
<tr>
<td>Paid</td>
<td>2.789</td>
<td>2.498</td>
<td>2.258</td>
<td>0.993</td>
<td>0.618</td>
</tr>
</tbody>
</table>

Test Accuracy (i.e., % Correctly Classified) = 44%

Penalized multinomial logistic regression: Coefficients (log-odds) fit on unpopular apps (N=150).
Optimal hyperparameter (inverse of regularization strength), $C = 2$.
Training accuracy = 53%. Cross-validation accuracy = 50% (standard deviation of 10%).

<table>
<thead>
<tr>
<th>Feature</th>
<th>Importance (mean impurity decrease)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality</td>
<td>0.385</td>
</tr>
<tr>
<td>PopularDeveloper</td>
<td>0.250</td>
</tr>
<tr>
<td>OtherVenue</td>
<td>0.180</td>
</tr>
<tr>
<td>Media</td>
<td>0.113</td>
</tr>
<tr>
<td>UserResources</td>
<td>0.073</td>
</tr>
</tbody>
</table>

Test Accuracy (i.e., % Correctly Classified) = 38%

Random Forest: Predictor importance ($n_{trees}$=100) analysis on unpopular apps (N$_{apps}$=150).
Optimal hyperparameters: minimum samples at leaf = 3, maximum depth = 5.
Training accuracy = 55%. Cross-validation accuracy = 48% (standard deviation of 10%).
TABLE 5: POPULAR PRODUCTS (BUNDLED V. FRAGMENTED FREEMIUM): MULTINOMIAL LOGIT AND RANDOM FOREST ANALYSES

<table>
<thead>
<tr>
<th>Revenue model</th>
<th>Media</th>
<th>Popular Developer</th>
<th>Other Venue</th>
<th>User Resources</th>
<th>Interrelated</th>
<th>Consumable</th>
<th>Quality</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1 (fragmented freemium=1)</td>
<td>-0.069</td>
<td>0.311</td>
<td>0.331</td>
<td>0.190</td>
<td>-0.440</td>
<td>2.070</td>
<td>0.160</td>
<td>100%</td>
</tr>
<tr>
<td>Model 2 (fragmented freemium=1)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-0.555</td>
<td>1.964</td>
<td>-</td>
<td>100%</td>
</tr>
</tbody>
</table>

Penalized multinomial logistic regression: Coefficients (log-odds) classifying bundled v. fragmented freemium revenue models for popular apps (N=100).

Optimal hyperparameter (inverse of regularization strength), $C = 1$.
Model 1: Training accuracy = 77%, Cross-validation accuracy = 73% (standard deviation of 15%).
Model 2: Training accuracy = 77%. Cross-validation accuracy = 78% (standard deviation of 15%).

<table>
<thead>
<tr>
<th>Feature</th>
<th>Importance (mean impurity decrease)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumable</td>
<td>0.398</td>
</tr>
<tr>
<td>Quality</td>
<td>0.181</td>
</tr>
<tr>
<td>Interrelated</td>
<td>0.105</td>
</tr>
<tr>
<td>PopularDeveloper</td>
<td>0.088</td>
</tr>
<tr>
<td>UserResources</td>
<td>0.082</td>
</tr>
<tr>
<td>OtherVenue</td>
<td>0.078</td>
</tr>
<tr>
<td>Media</td>
<td>0.068</td>
</tr>
</tbody>
</table>

Test Accuracy (i.e., % Correctly Classified) = 95%

Random Forest: Predictor importance ($n_{trees}=100$) analysis of bundled v. fragmented freemium revenue models for popular apps ($N_{apps}=100$).
Optimal hyperparameters: minimum samples at leaf = 1, maximum depth = 5.
Training accuracy = 85%. Cross-validation accuracy = 78% (standard deviation of 17%). Test accuracy = 95%.
<table>
<thead>
<tr>
<th>Method</th>
<th>Multi-case theory building</th>
<th>Machine learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Definition</strong></td>
<td>Process for using two or more cases (i.e., a rich empirical description of a specific situation) to build theory</td>
<td>Tools and process for modeling patterns in data</td>
</tr>
<tr>
<td><strong>Objectives</strong></td>
<td>Simple, accurate, robust and generalizable theory</td>
<td>Simple, accurate models for generalizable and robust prediction</td>
</tr>
<tr>
<td><strong>Guards against overfitting</strong></td>
<td>Replication logic</td>
<td>Cross-validation</td>
</tr>
<tr>
<td><strong>Guards against complexity</strong></td>
<td>Construct abstraction levels</td>
<td>Regularization techniques</td>
</tr>
<tr>
<td><strong>A priori assumptions</strong></td>
<td>Few</td>
<td>Few</td>
</tr>
<tr>
<td><strong>Sampling</strong></td>
<td>Theoretical</td>
<td>Random</td>
</tr>
<tr>
<td><strong>Model selection</strong></td>
<td>Researcher</td>
<td>Algorithm</td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td>Very small</td>
<td>Small to very large</td>
</tr>
<tr>
<td><strong>Strengths</strong></td>
<td>Empirically grounded</td>
<td>Empirically grounded</td>
</tr>
<tr>
<td></td>
<td>Rich description</td>
<td>Large scale</td>
</tr>
<tr>
<td></td>
<td>Identification of important theoretical constructs, relationships, and mechanisms</td>
<td>Precise identification of important constructs and patterns (e.g., equifinal paths, non-linearities, and effect sizes)</td>
</tr>
<tr>
<td><strong>Weaknesses</strong></td>
<td>Small scale</td>
<td>Cross-sectional</td>
</tr>
<tr>
<td></td>
<td>Researcher dependent</td>
<td>Atheoretical</td>
</tr>
<tr>
<td></td>
<td>Imprecise identification (Interactions, non-linearities, effect sizes)</td>
<td>Can be “black box” (e.g., neural networks)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No causal inference or significance test</td>
</tr>
</tbody>
</table>
FIGURE 1: MAJOR REVENUE MODEL FREQUENCY – POPULAR (TOP) V. UNPOPULAR (BOTTOM) (N = 66,652).
FIGURE 2: TOP THREE AND BOTTOM THREE MARKETS (% OF POPULAR APP PRODUCTS) BY REVENUE MODEL.
FIGURE 3: REVENUE MODEL (% PRODUCTS BY MARKET) – POPULAR V. UNPOPULAR (Table A1 for all categories).
FIGURE 4: FREQUENCY OF #UP-SELL PRODUCTS (FREEMIUM)

Bimodal distribution reveals TWO distinct freemium revenue models: Bundled and Fragmented.
FIGURE 5: DECISION TREE (POPULAR PRODUCTS): Classifies Revenue Models

Test Accuracy (i.e., % Correctly Classified) = 68%

Optimal hyperparameters: minimum samples at leaf = 12, maximum depth = 3.
Training accuracy = 65%. Cross-validation accuracy = 64% (standard deviation of 9%).
Free is third-party only.
**FIGURE 6:** DECISION TREE (UNPOPULAR PRODUCTS): Classifies Revenue Models

**Test Accuracy (i.e., % Correctly Classified)** = 44%

Optimal hyperparameters: minimum samples at leaf = 3, maximum depth = 3. Training accuracy = 53%.

Cross-validation accuracy = 49% (standard deviation of 9%).

Free is third-party only.
FIGURE 7: DECISION TREE (POPULAR PRODUCTS): Classifies bundled v. fragmented freemium revenue models

Test Accuracy (i.e., % Correctly Classified) = 100%
Optimal hyperparameters: minimum samples at leaf = 1, maximum depth = 1.
Training accuracy = 77%. Cross-validation accuracy = 78% (standard deviation of 15%).
APPENDIX – MACHINE LEARNING ANALYSIS

Measures

We rely on a number of binary variables in order to scale our machine learning analysis for easy interpretation and to avoid over-fitting (Varian, 2014). Specifically, we measure quality signal with three binary variables. Media is a binary variable = 1 (otherwise 0) if the product is referenced in any media reports including awards within six months of its launch on the App Store. To collect this data, we conducted extensive online searches for the app within this 6-month time frame. We used Amazon Mechanical Turk as a robustness check – i.e., we employed two workers per app to locate media reports, and verified consistency among each of the Mechanical Turk workers and our own collected data. In the case of disagreements, we re-examined each of the media references for dates and other details.

Our second measure of a quality signal is OtherVenue. This binary variable = 1 (otherwise 0) if the focal app has a closely related product that already exists in at least one other domain (e.g., print). We followed a similar procedure to the above (i.e., our own online search, Mechanical Turk) to check whether the product or related products existed in other domains prior to launch on the App Store.

Our third measure of a quality signal is PopularDeveloper. This binary variable = 1 if the developer of the focal app has other apps that are also listed as popular. If the developer has no such apps, then the variable = 0.

We measure product quality with a single variable, Quality, that is the average of all user ratings (out of five stars). We obtain these data from the App Store page of the focal app. Prior work finds that the average of user ratings is an accurate indicator of underlying quality, but does not directly affect performance (Duan et al., 2008). To avoid over-fitting, we round quality to the
nearest integer. We measure whether the app is a sales channel with a single binary variable, Transaction. This variable = 1 (otherwise 0) if the app is a conduit to sales of a non-app (e.g., physical) product. Example apps include Domino’s Pizza and Target.

We measure whether the product uses a third-party free revenue model with the binary variable ThirdParty. ThirdParty = 1 (otherwise 0) if the product is free (i.e., up-front price is zero), offers no up-sells, and is not a sales channel (i.e., Transaction = 0). We conducted additional online searches to confirm that third-parties are the primary source of revenues, if any.

We measure user resources with a single variable, UserResources. This binary variable = 1 (otherwise 0) if the product collects information or other resources (e.g., financial assets) about users that is relevant for potential third-party payers. Specifically, we coded data on user preferences (e.g., OpenTable), friends and activities (e.g., Facebook), specific problems (e.g., WebMD), and spending habits (e.g., Mint Personal Finances & Money) as user resources. We adopt a binary measure in order to provide a conservative measure that is robust and does not require assessing the specific value of different user resources.

For freemium apps, we measure up-sells by the variables, Interrelated and Consumable. To code the Interrelated measure, we assess whether the value of one up-sell feature improves with the presence of another feature and vice versa (i.e., super-modularity). We downloaded the apps to evaluate features and their interrelatedness. We code this binary variable =1 if the features are interrelated. The binary Consumable measure =1 if upon purchase, the up-sell is used and then depleted. This is consistent with Apple’s definition.\textsuperscript{12} Examples include extra lives in games or tokens for virtual purchases.

\textsuperscript{12} See https://developer.apple.com/in-app-purchase/.
We measure subscription by whether the app offers Subscriptions = 1 (otherwise 0). A subscription provides time-limited access to an app (e.g., one month), and so is usually set up as a recurring payment.

In sum, we collect additional measures for the ML analyses: Media, OtherVenue, PopularDeveloper, UserResources, Quality, Interrelated, Consumable, ThirdParty, Transaction, and Subscription. We originally tracked 676 apps. Of those 676, we filter out products that use the Transaction revenue model as these relate to sales channels, not meaningful products. This leaves us with 202 popular and 202 unpopular apps with balance across the revenue models. Summary statistics are presented in Table A2. We also randomly selected 50 popular and 50 unpopular apps (randomized across all revenue models and excluding Transaction revenue model) to serve as test sets.

Methods

We employ a machine learning framework. We split our analyses first by popular and unpopular products, and then conduct a full (joint) analysis. Analysis of each segment follows the same basic structure (Choudhury et al., 2018): (1) problem formulation (choosing a machine learning model, relevant features, and a loss function), (2) k-fold cross-validation\(^{13}\) to find optimal hyperparameters,\(^{14}\) (3) model testing on out-of-sample data to evaluate performance, (4) repeating steps 1-3 for different machine learning models, (5) interpretation of results to add or extend our insights.

\(^{13}\) k-fold cross-validation involves splitting the data into k subgroups, training (i.e., “fitting”) the model on k-1 of those sub-groups and then testing on the remaining k\(^{th}\) group. Repeating this process k times and averaging the k test scores provides an estimate for out-of-sample model performance. As is common, we set k=10 throughout our study.

\(^{14}\) Hyperparameters are input parameters that are not “learned,” but rather define the problem. For example, the degree of large coefficient penalization or number of leaves on a decision tree are regularization hyperparameters. “Optimal hyperparameters” are the hyperparameters which result in the best model performance, evaluated using cross-validation.
Problem formulation

We model the choice of revenue model as a multiclass classification problem. That is, we seek to predict the revenue model (free, freemium, paid) based on the relevant features. In this formulation, we combine bundled and fragmented freemium revenue models into the broader freemium model. Subsequent analysis aims to predict the choice of bundled or fragmented models conditional on the choice of freemium revenue model. As is common practice (Choudhury et al., 2018), we report training, cross-validation, and test set accuracies (i.e., % correctly classified) for each model.

In the first revenue model classification step, we include five features: Media, OtherVenue, PopularDeveloper, UserResources, Quality. In the second freemium revenue model classification (i.e., bundled vs. fragmented freemium), we run both a full model (with the five features: Media, OtherVenue, PopularDeveloper, UserResources, Quality) and with two measures: Interrelated and Consumable. We do so because our multiple-cases highlight that the Interrelated and Consumable features are most predictive.

We begin with a penalized multinomial logistic regression classifier, with a cross-entropy objective function. In a machine learning extension to this traditional approach, we include an $L2$ (i.e., squared) regularization term in our objective function. This function imposes a penalty that is proportional to the square of the fit coefficients. By penalizing large coefficients, the penalized multinomial logistic regression will produce more conservative coefficient estimates, and so is expected to perform better on out-of-sample test data. Regularization complements the

---

15 We note that in order to retain balance between the classes (free, freemium, paid), we use down-sampling – i.e., random sampling of 50 freemium apps from the 100 total (since we collected 50 bundled and 50 fragmented freemium apps). Thus, our initial classification (between free, freemium, and paid revenue models) consists of 150 apps for each popular and unpopular apps.
16 We train our classifier on 100 apps (for each popular and unpopular), where we have balance across the two classes (bundled and fragmented).
traditional (i.e., non-penalized) approach by highlighting the features of particular importance, since these will remain large even with penalization. To tune the degree of penalty, we include the hyperparameter, $C$, which is the inverse of regularization strength (smaller $C$ means greater penalty). We conduct cross-validation to find the hyperparameter (i.e., $C$) that results in the highest performing classifier.

We use two additional machine learning algorithms. The first is a Decision Tree classifier. Decision trees are based on a greedy heuristic algorithm used to iteratively add decision nodes that maximize a local objective.\footnote{The problem of finding the optimal decision tree is NP-complete, and hence a local heuristic (e.g., maximizing information gain) is needed.} In our case, we choose entropy as a measure of information gain (i.e., Kullback-Leibler divergence; Kullback and Leibler, 1951), which is maximized for each choice of decision criteria (i.e., node). Decision trees are particularly effective for highlighting multiple paths through the data, and elucidating simple often non-linear relationships (Varian, 2014). In this setting, we characterize decision trees with two hyperparameters: minimum number of samples at a leaf and maximum tree depth. We conduct grid search (using cross-validation) to find the hyperparameter pair that define the highest-performing decision tree.

We also use a Random Forest ensemble method. This method generates many decision trees (in our case, $n=100$) on sub-samples of the data (chosen with replacement to match the size of the original data set). To avoid correlation between trees, only a (random) subsample of features are considered at each decision node (i.e., to build each new sub-tree). As is common, we set the number of selected features approximately equal to the square-root of the total number of features (James et al., 2013). The resulting decision trees are averaged in order to make predictions. Random forests are highly effective for measuring relative feature importance, since
these are based on many uncorrelated decision trees. It is for the same reason, however, that random forests are less interpretable – any one tree only contributes a fractional weight to final predictions. Random forests can also be characterized by two hyperparameters – the minimum number of samples at a leaf and maximum tree depth. The hyperparameters are similarly chosen using grid search and cross-validation.

**Results: Popular apps**

We report that the *penalized multinomial logistic regression* model achieves a 10-fold cross-validation accuracy of 64% (with validation score standard deviation 7%), and a test accuracy of 68% (Table 3). This improves on the naïve guess of 33% (i.e., one of three revenue models). The optimal hyperparameter – the inverse of regularization strength – is approximately equal to one, which suggests a high degree of regularization. In results available from the authors, we also compare the traditional multinomial logistic regression model with the regularized one, and find smaller coefficient estimates in the penalized model (as expected since this is a direct effect of regularization), but consistent relative effect sizes and directions of coefficient estimates and relative effect sizes.

We use cross-validation to find that the best-performing *decision tree* has a minimum of twelve samples per leaf and maximum depth of three (Figure 5). This tree achieves 64% accuracy on 10-fold cross-validation (9% standard deviation), and an out-of-sample test accuracy of 68% that substantially improves on the 33% naïve guess.

Using the random forest method to extend the decision tree analysis, we find that a minimum of ten samples per leaf and maximum depth of four are the highest-performing hyperparameters. The additional depth of the tree (which risks overfitting) is balanced by the effect of averaging across many trees ($n=100$). The random forest achieves a test accuracy of
68%. We report relative feature importance in terms of the “mean impurity decrease” (Breiman et al., 1984)\textsuperscript{18} (Table 3).

We also predict the choice of bundled or fragmented revenue model \textit{given} the choice of freemium revenue model. For \textit{penalized logistic regression}, we compare two models: one fully specified (i.e., \textit{Media, OtherVenue, PopularDeveloper, Rating, UserResources, Interrelated, Consumable}) and one with only the constructs identified in the propositions (i.e., \textit{Interrelated, Consumable}). The results are in in Table 4. In both cases, we find the optimal regularization hyperparameter, $C=1$.

The \textit{decision tree} for predicting bundled or fragmented freemium revenue models (given the choice of freemium revenue model) has a minimum of one sample per leaf and a maximum depth of one. It achieves a test accuracy of 100%. The tree is shown in Figure 6.

Finally, we run a \textit{random forest} to extend our decision tree for predicting the type of freemium revenue model. Using cross-validation, we find the optimal hyperparameters to be a minimum of one sample per leaf, and a maximum depth of five. The random forest achieves a training accuracy of 85% and a test accuracy of 95%. Feature importance is reported in Table 4.

\textbf{Results: Unpopular apps}

We conduct the same analysis on unpopular apps. We consistently find significant difference in performance of the machine learning analyses. In predicting the choice of revenue model we first report results from the \textit{penalized multinomial logistic regression} model. We find a cross-validation accuracy of 50% (with validation score standard deviation 10%), and a test accuracy of 44% (Table 5). This slightly improves on the naïve guess of 33\% (i.e., one of three

\textsuperscript{18}“Mean impurity decrease” is the average (across the trees in the forest) decrease in node impurity (i.e., node mis-classifications), weighted by the probability of reaching that node. The probability of reaching a node is approximated by the proportion of samples that reach the given node.
revenue models). The optimal hyperparameter – the inverse of regularization strength – is approximately equal to one, which suggests a high degree of regularization. In results available from the authors, we compare the penalized multinomial logistic regression to the traditional full model with no regularization. As expected, we find that the penalized logistic regression provides smaller coefficient estimates than with no penalty.

Using cross-validation we find that the best-performing decision tree has a minimum of three samples per leaf and maximum depth of three (Figure 7). This tree achieves 49% accuracy on 10-fold cross-validation (9% standard deviation), and lower out-of-sample test accuracy of 44%.

Using the random forest method, we find that a minimum of three samples per leaf and maximum depth of five are the highest-performing hyperparameters. As in the popular app analysis, the random forest has a greater maximum depth than the single decision tree. The risk of overfitting is balanced by the effect of averaging across many trees \( n=100 \). The random forest achieves a test accuracy of 38%, barely higher than a naïve guess (33%). We report relative feature importance (Table 5).

We note that since the accuracy is low, results should be interpreted with caution. In results available from the authors, we also use our popular app classification models to predict the revenue models of unpopular apps. That is, we test our popular app models on the unpopular apps. We find a substantial decline in accuracy. The popular app classifiers (penalized multinomial logistic regression, decision tree, random forest) achieve an accuracy of approximately 47% when test on the unpopular apps. This suggests that unpopular apps are choosing revenue models differently (i.e., with different variable configurations) to popular apps.

65
In results available from the authors, we also predict the choice of bundled or fragmented revenue model given the choice of freemium revenue model for unpopular apps. For penalized logistic regression, we compare two models: one fully specified (i.e., Media, OtherVenue, PopularDeveloper, Rating, UserResources, Interrelated, Consumable) and one with only the constructs identified in the propositions (i.e., Interrelated, Consumable). In both cases, we find the optimal regularization hyperparameter, $C=1$. The fully specified model achieves a test accuracy of 62%, while the latter model achieves a test accuracy of 85%.

The decision tree for predicting bundled or fragmented freemium revenue models (given the choice of freemium revenue model) has a minimum of one sample per leaf and a maximum depth of two. It achieves a test accuracy of 62%.

Finally, we run a random forest for predicting the type of freemium revenue model for unpopular apps. Using cross-validation, we find the optimal hyperparameters to be a minimum of sixteen sample per leaf, and a maximum depth of four. The random forest achieves a test accuracy of 85%. Feature importance are available from the authors.

REFERENCES

<table>
<thead>
<tr>
<th>Market Category</th>
<th>Popular</th>
<th>Unpopular</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Paid</td>
<td>Free</td>
</tr>
<tr>
<td>Finance</td>
<td>2.50%</td>
<td>82.08%</td>
</tr>
<tr>
<td>Lifestyle</td>
<td>0.00%</td>
<td>79.57%</td>
</tr>
<tr>
<td>News</td>
<td>4.09%</td>
<td>79.55%</td>
</tr>
<tr>
<td>Travel</td>
<td>2.35%</td>
<td>77.93%</td>
</tr>
<tr>
<td>Food &amp; Drink</td>
<td>6.61%</td>
<td>73.13%</td>
</tr>
<tr>
<td>Business</td>
<td>7.14%</td>
<td>70.59%</td>
</tr>
<tr>
<td>Sports</td>
<td>2.92%</td>
<td>67.08%</td>
</tr>
<tr>
<td>Catalogs</td>
<td>3.33%</td>
<td>63.33%</td>
</tr>
<tr>
<td>Weather</td>
<td>13.10%</td>
<td>60.70%</td>
</tr>
<tr>
<td>Navigation</td>
<td>8.33%</td>
<td>60.42%</td>
</tr>
<tr>
<td>Entertainment</td>
<td>1.26%</td>
<td>60.25%</td>
</tr>
<tr>
<td>Utilities</td>
<td>2.50%</td>
<td>56.25%</td>
</tr>
<tr>
<td>Medical</td>
<td>6.67%</td>
<td>54.58%</td>
</tr>
<tr>
<td>Reference</td>
<td>3.75%</td>
<td>53.33%</td>
</tr>
<tr>
<td>Music</td>
<td>5.00%</td>
<td>50.42%</td>
</tr>
<tr>
<td>Health &amp; Fitness</td>
<td>9.01%</td>
<td>45.49%</td>
</tr>
<tr>
<td>Social Networking</td>
<td>2.08%</td>
<td>45.42%</td>
</tr>
<tr>
<td>Productivity</td>
<td>4.58%</td>
<td>40.00%</td>
</tr>
<tr>
<td>Education</td>
<td>4.17%</td>
<td>39.17%</td>
</tr>
<tr>
<td>Books</td>
<td>9.24%</td>
<td>33.19%</td>
</tr>
<tr>
<td>Photo &amp; Video</td>
<td>4.17%</td>
<td>31.25%</td>
</tr>
<tr>
<td>Magazines &amp; Newspapers</td>
<td>0.96%</td>
<td>22.55%</td>
</tr>
<tr>
<td>Games</td>
<td>4.97%</td>
<td>19.38%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>3.32%</strong></td>
<td><strong>35.16%</strong></td>
</tr>
</tbody>
</table>
### TABLE A2: SUMMARY STATISTICS FOR ML ANALYSES (N=404).

<table>
<thead>
<tr>
<th>Revenue model</th>
<th>Media</th>
<th>Popular Developer</th>
<th>Other Venue</th>
<th>User Resources</th>
<th>Interrelated</th>
<th>Consumable</th>
<th>Subscription</th>
<th>Quality</th>
<th># Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREEMIUM BUNDLED</td>
<td>0.58</td>
<td>0.53</td>
<td>0.33</td>
<td>0.16</td>
<td>0.57</td>
<td>0.18</td>
<td>0.47</td>
<td>3.80</td>
<td>51</td>
</tr>
<tr>
<td>FREEMIUM FRAGMENTED</td>
<td>0.66</td>
<td>0.62</td>
<td>0.24</td>
<td>0.18</td>
<td>0.34</td>
<td>0.72</td>
<td>0.42</td>
<td>3.76</td>
<td>50</td>
</tr>
<tr>
<td>FREE (THIRD-PARTY)</td>
<td>0.90</td>
<td>0.53</td>
<td>0.41</td>
<td>0.78</td>
<td>0.00</td>
<td>0.02</td>
<td>0.00</td>
<td>3.82</td>
<td>51</td>
</tr>
<tr>
<td>PAID</td>
<td>0.84</td>
<td>0.34</td>
<td>0.18</td>
<td>0.12</td>
<td>0.02</td>
<td>0.18</td>
<td>0.04</td>
<td>3.84</td>
<td>50</td>
</tr>
</tbody>
</table>

Summary statistics for popular apps.

<table>
<thead>
<tr>
<th>Revenue model</th>
<th>Media</th>
<th>Popular Developer</th>
<th>Other Venue</th>
<th>User Resources</th>
<th>Interrelated</th>
<th>Consumable</th>
<th>Subscription</th>
<th>Quality</th>
<th># Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREEMIUM BUNDLED</td>
<td>0.08</td>
<td>0.20</td>
<td>0.12</td>
<td>0.02</td>
<td>0.18</td>
<td>0.72</td>
<td>0.12</td>
<td>0.70</td>
<td>50</td>
</tr>
<tr>
<td>FREEMIUM FRAGMENTED</td>
<td>0.02</td>
<td>0.26</td>
<td>0.00</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2.14</td>
<td>52</td>
</tr>
<tr>
<td>FREE (THIRD-PARTY)</td>
<td>0.16</td>
<td>0.00</td>
<td>0.14</td>
<td>0.10</td>
<td>0.00</td>
<td>0.04</td>
<td>0.00</td>
<td>0.38</td>
<td>50</td>
</tr>
<tr>
<td>PAID</td>
<td>0.02</td>
<td>0.08</td>
<td>0.02</td>
<td>0.10</td>
<td>0.28</td>
<td>0.26</td>
<td>0.06</td>
<td>1.00</td>
<td>50</td>
</tr>
</tbody>
</table>

Summary statistics for unpopular apps.

### TABLE A3: FREEMIUM V. PAID REVENUE MODELS: “No UserResources and PopularDeveloper” decision tree node (Figure 5).

<table>
<thead>
<tr>
<th>Revenue model</th>
<th>Media</th>
<th>Other Venue</th>
<th>User Resources</th>
<th>Mean Rating</th>
<th>Developer Popularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREEMIUM</td>
<td>0.800</td>
<td>0.160</td>
<td>0.000</td>
<td>3.960</td>
<td>1.00</td>
</tr>
<tr>
<td>PAID</td>
<td>0.813</td>
<td>0.188</td>
<td>0.000</td>
<td>3.938</td>
<td>1.00</td>
</tr>
</tbody>
</table>