WILL THEY PAY?

PREDICTING IF USERS WILL PAY FOR APPS USING WORLDWIDE MOBILE APP USER BEHAVIOR DATA

Jaysmit Jadhav
jaysmit@mail.com

ABSTRACT

This project proposes, evaluates and interprets machine learning models to predict if mobile phone users will pay for apps. We used the Worldwide Mobile App User Behavior dataset which is obtained by surveying 10208 people from more than 15 countries on their mobile app usage behavior. We first explored the data to understand the user demographics, personalities and their usage patterns. We then predicted the user payment behavior using different classification and ensemble algorithms. Random Forest and Extra Trees classifier were the most successful models achieving validation and testing accuracies of 70%. We further use model interpretability libraries SHAP, Eli5 and Permutation Probability to understand the predictions and the feature values responsible.

I INTRODUCTION

Since the launch of modern smartphone in 2007, the mobile phone industry has proliferated. Over 3 billion people spend substantial amounts of their day using apps on their smartphones with global internet penetration standing at 57%. Today, there are almost 2.46 million apps available for download in Google Playstore, 1.96 million in Apple App Store, 700,000 in Windows Store, and 479,000 in Amazon Appstore. Although many have seen mammoth success, most of them turn out to be unsuccessful. Of the paid apps, about 90 percent are downloaded less than 500 times per day — and earn less than $1,250 a day. Moreover, 80% of them received less than 100 downloads. While competition in the app market is high, failure isn't always a result of getting lost in the noise. In most cases, there are other contributing factors. Poorly researched market and audience would most likely top that list.

This work is relevant not only relevant to B2C paid apps but all apps. The analysis of the results will help facilitate new challenges to market-driven software engineering related to packaging requirements, feature space, quality expectations, app store dependency, price sensitivity and ecosystem effect.

II DATASET

We used Lim, Soo Ling, 2014, "Worldwide Mobile App User Behavior Dataset", https://doi.org/10.7910/DVN/27459, Harvard Dataverse, V1 dataset for our project. The dataset contains the results of a survey of 10208 people from more than 15 countries on their mobile app usage behavior. The countries include USA, China, Japan, Germany, France, Brazil, UK, Italy, Russia, India, Canada, Spain, Australia, Mexico, and South Korea. The respondents were asked about: (1) their mobile app user behavior in terms of mobile app usage, including the app stores they use, what triggers them to look for apps, why they download apps, why they abandon apps, and the types of apps they download. (2) their demographics including gender, age, marital status, nationality, country of residence, first language, ethnicity, education level, occupation, and household income (3) their personality using the Big-Five personality traits.

The dataset also contains metadata such as os, browser, screen resolution, flash version, etc. from the machine the users used to take the survey.

III DATA WRANGLING

The dataset comprises of 10208 observations and 161 variables. 71.5% of the data contains missing values with some variables having almost 98% values missing. The 161 variables were distributed as 45 numeric, 13 categorical, 101 boolean and 2 date. Some variables have very high cardinality. Model variable has 4421 distinct values.

The survey questionnaire was used as a guide to change the numeric columns to correct categorical columns. The high cardinality columns were mostly as a result of string values inputted by the users. String manipulation techniques were used to convert strings to categories. The missing values from boolean columns were filled with zeros after studying the dataset.

The target variable 'do not pay for apps' was one of the 10 multiple choice options for the survey question – Why do you spend money on an app? In order to minimize bias, the other 9 options were dropped from the data. The metadata was excluded for modeling in order to restrict the predictions strictly based on user behavior. The cleaned data was saved as a pickle file to retain the changed datatypes.

IV EXPLORATORY DATA ANALYSIS

In this section, we investigated and explored the data to understand the user demographics, personalities, user adoption of the app store, app needs and the rationale for selecting or abandoning an app. We also focus on the differences of these findings between the payment behaviors.

a. Distribution of users

The demographics of the surveyed people are explored in this subsection.

a.1. Gender Ratio

The data has a balanced gender ratio of 48.63% males and 51.73% females. The number of men who paid for apps is almost same as men who didn't. But there is a larger gap in this ratio for women.

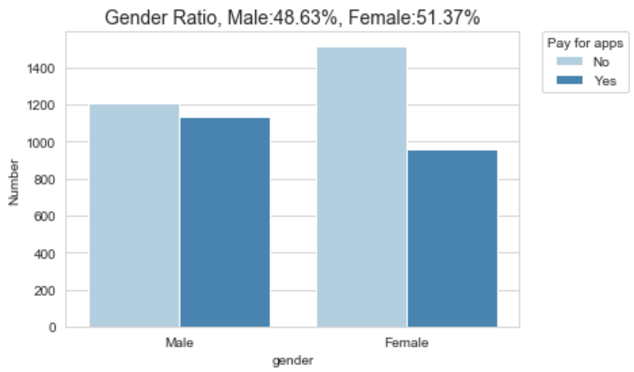

Figure 1 Gender Ratio and payment behavior

a.2. Nationality

The number of users who pay for apps is more than twice the number of users who don't in China. Mexico is the only other country where more number of users pay for apps than don't.

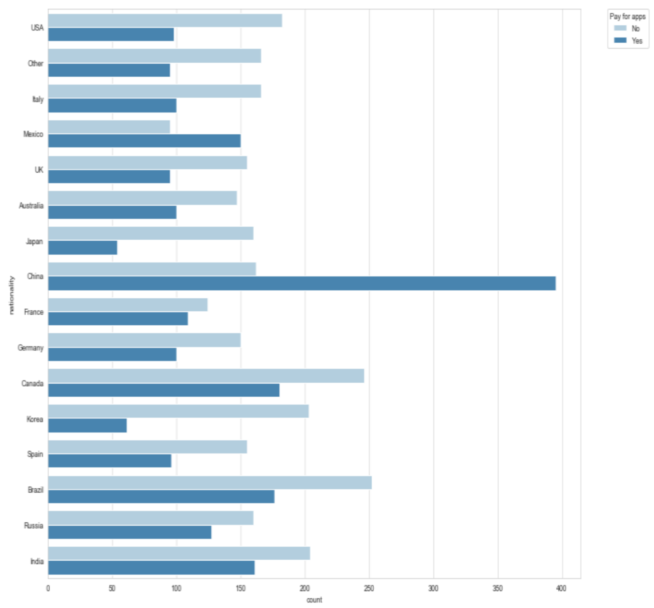

Figure 2 Nationality and Payment Behavior

a.3. Age

Age has a right-skewed distribution with peak at 20.

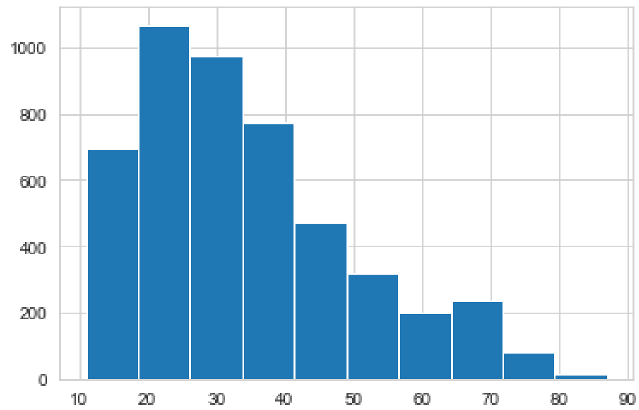

Figure 3 Age

b. Personality Traits

Personality may influence the types of apps one likes and if they are willing to pay for an app. In this section, we explored the different personality traits on a scale of 1 to 7, value 1 is where they 'Strongly disagree' with a characteristic and 7 as 'Strongly agree'.

We see influence of social desirability bias for all the personality traits' distributions. For all traits, the most frequent rating was in between 2-6 which suggests that people shy away from rating themselves with strong characteristics. Traits with positive connotation such as extraverted, enthusiastic, dependable, self-disciplined, open to new experiences, sympathetic, warm, calm, emotionally stable showed left-skewed distribution i.e. majority of people agree that these traits apply to them. Traits with negative connotation such as critical, quarrelsome, anxious, easily upset, disorganized, careless, conventional, uncreative showed right-skewed distribution i.e. majority of people disagree that these traits apply to them. Reserved, quiet was the only trait which showed almost a normal distribution.

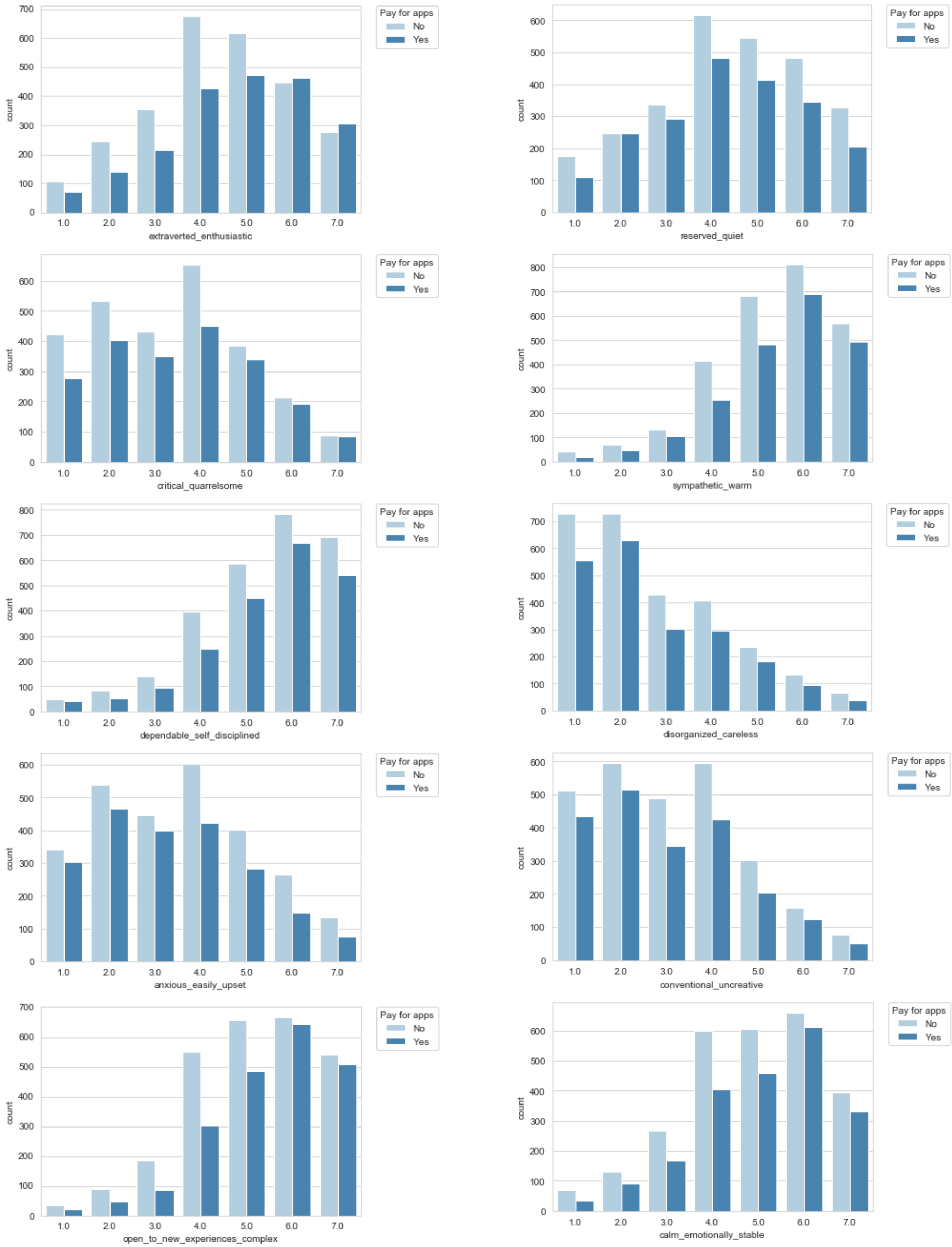

Figure 4 Personality Traits showing influence of social desirability bias. Users who strongly agree that they are extroverted and enthusiastic were the only group where there were more payers than non-payers.

c. App Store Adoption

It is important to understand how best to develop apps and app stores such that users can find apps. In this section we investigate user behavior relating to seeking apps, in terms of the platform used, frequency of use of that platform, frequency of downloads and methods used to search for apps.

c.1. User distribution across mobile app platforms

Google (Android) Play Store was the most used app store followed by Apple (iOS) App Store. This is consistent with the current market share of smartphone operating systems.

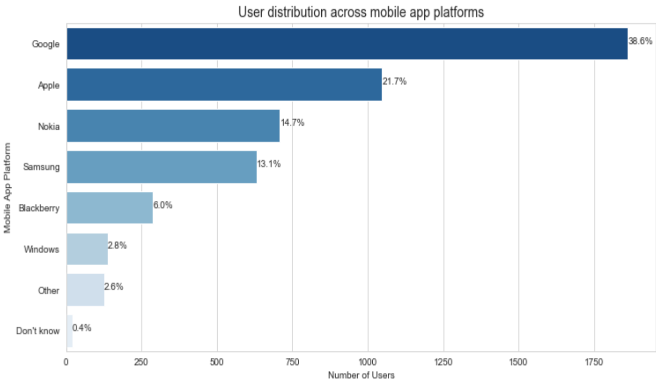

Figure 5 User distribution across mobile app platforms

c.2. Frequency of visiting app store

More than one a week was the most common frequency that users visited their app store. This was followed by less than once a month and once a week. The least common frequency of visiting the app store was several times a day. Approximately, 9% of users reported not visiting the app stores to look for apps.

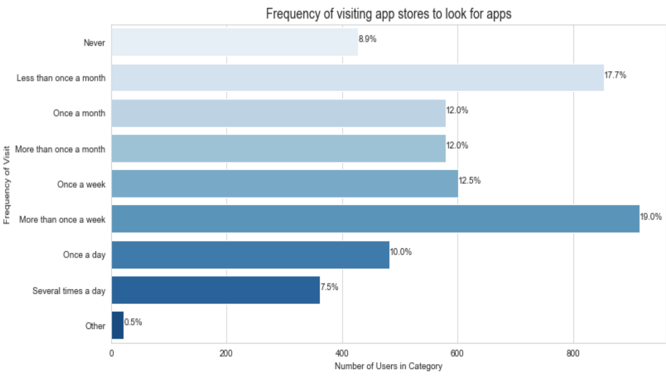

Figure 6 Frequency of visiting app stores to look for apps

c.3. Average Apps downloaded per month

The highest proportion of users downloaded 2–5 apps per month (40%). This was followed by 0–1 apps (35%), 6–10 apps (14%), 11–20 apps (7%), and 21–30 apps (2%). Only 2% of users downloaded more than 30 apps per month.

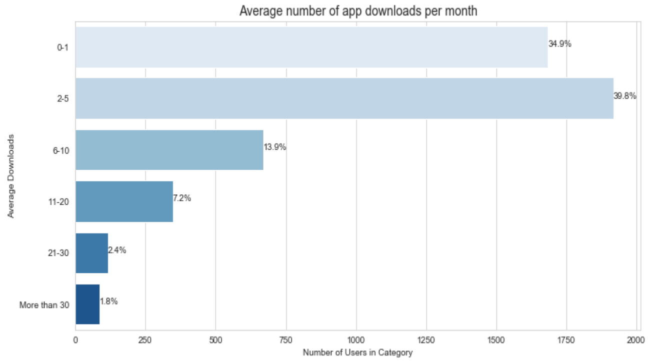

Figure 7 Average number of app downloads per month

c.4. Methods used to find apps

The majority of people found apps by keyword search in the app store (19%). This was followed by browsing randomly (17%), using search engines such as Google (16%), looking at top downloads chart (13%), and comparing several apps(11%). The least number of users reported downloading the first app they found(3%), suggesting that users tend to spend some time choosing apps, even if the apps were free.

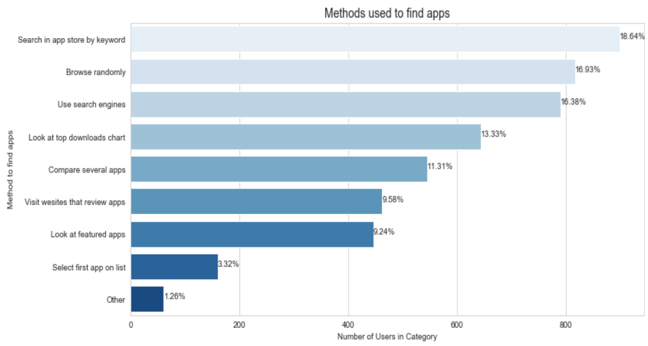

Figure 8 Methods used to find apps

d. User Needs

In addition to the mechanics of finding apps, there are the fundamental needs of the users. In this section, we aim to understand what might prompt a user to consider looking for an app in the first place, why they download apps, and which types of apps they prefer.

d.1. Triggers

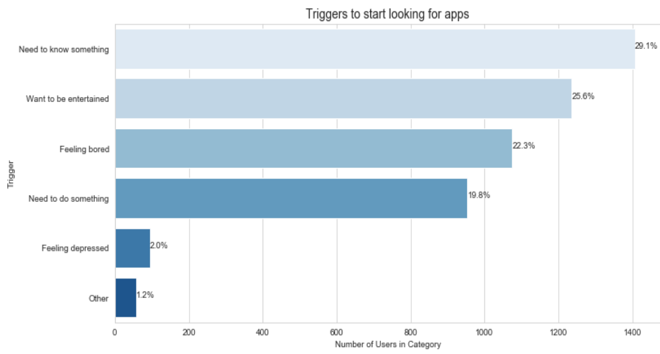

Figure 9 Triggers to start looking for apps

The most popular situation that triggered users to look for apps was when they needed to know something

(29%), followed by when they wanted to be entertained (26%), and when they were feeling bored (22%). The least popular reason to look for apps was when users were depressed (2%). However, the respondents' willingness to specify this option might have been influenced by social desirability bias.

d.2. Reasons for downloading apps

The most popular reason for users to download an app was to be entertained (20%), followed by to carry out a task (19%). The third most popular reason for users to download an app was because the app was recommended by friends or family (11%). This shows the importance of viral marketing and social networks on app downloads. Curiosity was also an important reason (10%), which meant that novel or quirky apps have the potential to attract downloads in the app store.

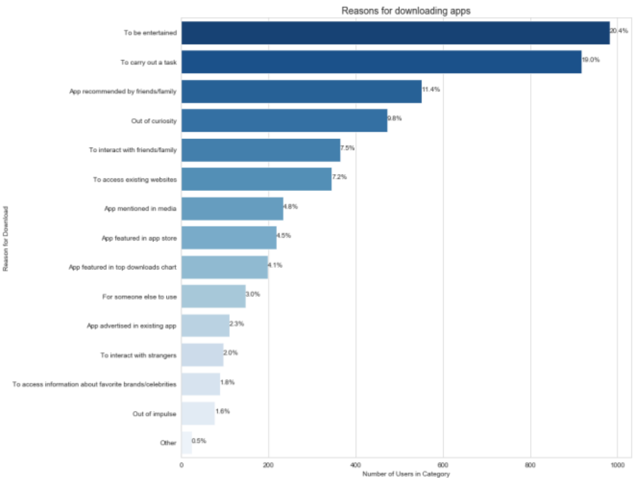

Figure 10 Reasons for downloading apps

d.3. Types of apps

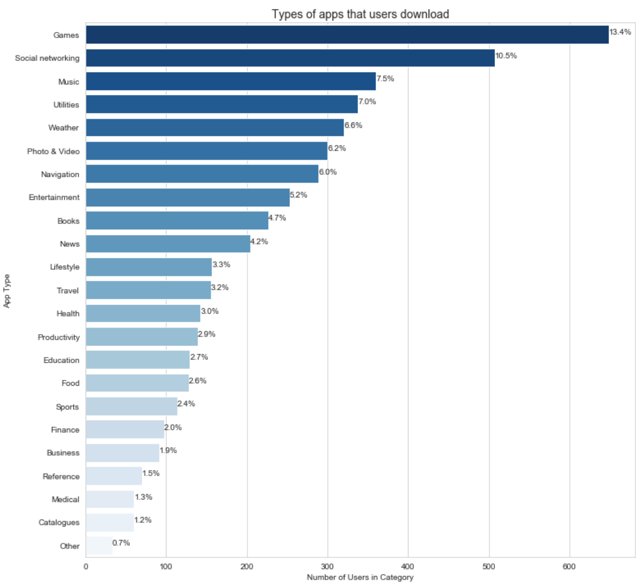

Figure 11 Types of apps that users download

The most popular app category was games (13%) followed by social networking (11%) and music apps (8%), which is consistent with the fact that the most common reason to download apps was to be entertained. Utility apps, weather apps and navigation apps were very popular too, indicating that apps play an important role in supporting very specific tasks and providing specific information.

e. Influencing features for selection or abandonment of apps

Apps must be advertised through app stores, potentially making non-functional and packaging requirements as important as functional requirements. In this section, we investigate the importance of app features versus descriptions, ratings, price, and perceived quality.

e.1. Factors that influence users' choices of apps

The most important factors that people consider when choosing apps were: price (17%), app features (14%), app description (13%), reviews by other users (12%), and star ratings (11%). The least important factor that influenced a user's choice of apps was the developer (2%). This meant that developers would find it difficult to use the success of their previous apps to promote future apps.

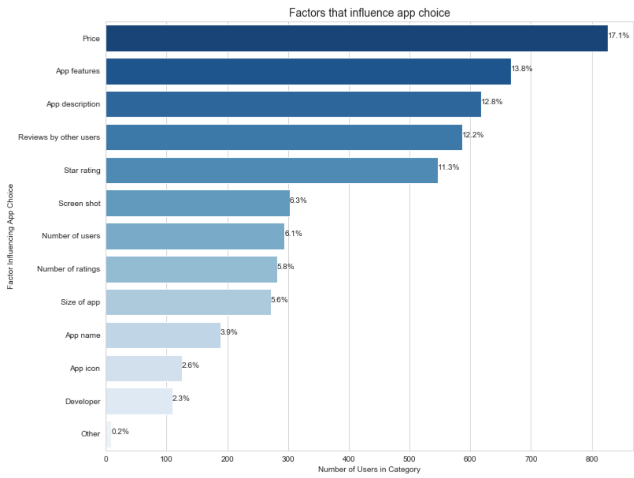

Figure 12 Factors that influence app choice

e.2. Reasons for rating apps

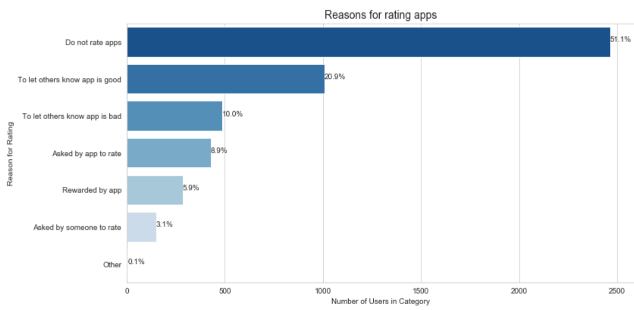

Figure 13 Reasons for rating apps

51% of users did not rate apps. The most popular reasons for rating apps was to let other users know that the app was good (21%), followed by to let other users know that the app was bad (10%). Interestingly, the app rewarding users to rate it (6%) was a less popular reason compared to the app simply reminding the users to rate it (9%).

e.3. Reasons for paying for apps

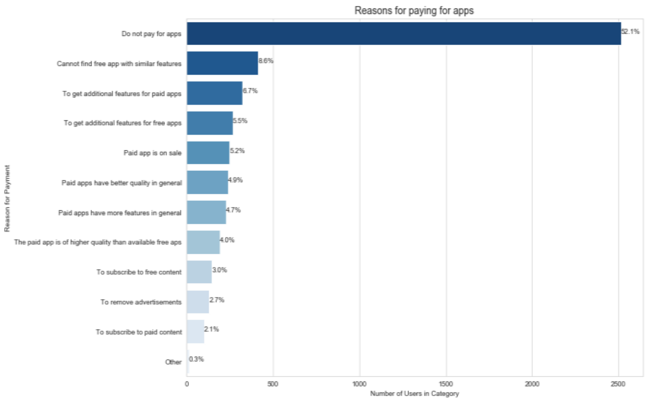

Figure 14 Reasons for paying for apps

Most app users did not pay for apps (52%). The most popular reasons to pay for apps were that users could not find free apps with similar features (9%). This was followed by the need to get additional features for paid apps (7%) and for free apps (6%), and that the apps were on sale (5%). Not many people paid to remove advertisements (3%). The least common reason people paid for apps was to subscribe for paid content (2%). This might be that when the content had to be paid for, users expected the app to be free.

e.4. Reasons for abandoning apps

The most common reason for app users to abandon an app was because they did not need the app anymore (14%). This was followed by finding better alternatives (11%) and getting bored of the app (10%). This finding suggested that many apps served temporary functions, unlike desktop software. Non-functional requirements such as performance, reliability and usability, were important for app users. Reasons such as the app crashed, the app did not have the required features, the app was too slow, the app was difficult to use, and the app did not work, were, on average, adequate reasons for more than 37% of users for abandoning an app. This result showed that the quality of an app was crucial to encourage continued usage. Only 4% of users stopped using an app because it invaded their privacy. However, this might be due to app users being largely unaware of their privacy being invaded and the implications.

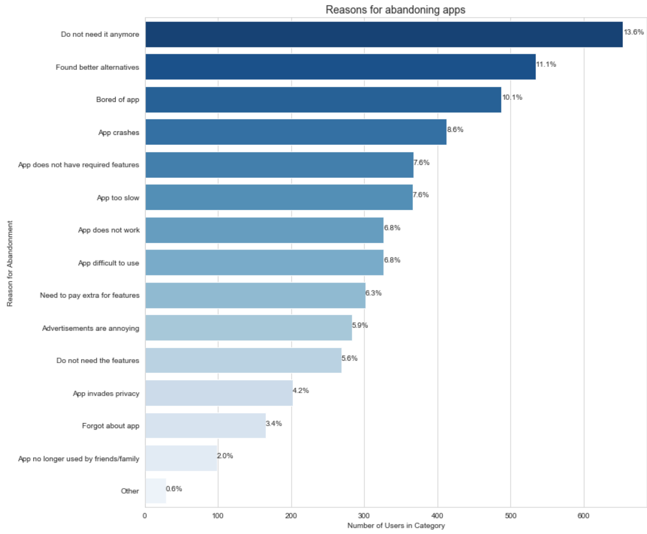

Figure 15 Reasons for abandoning apps

V METHODS

A variety of machine learning algorithms were used to perform binary classification of the target variable 'do_not_pay_for_apps'. The following learning algorithms were implemented using scikit-learn library.

a. Logistic Regression

In logistic regression, the predicted probability of not paying for apps is of the form:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

Where an affine transformation, $\theta^T x$ of the input user and features x are mapped onto the probability space by the sigmoid function. To fit the parameters θ of the model, log-likelihood $l(\theta)$ is maximized.

$$l(\theta) = \sum_{i=1}^{n} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))$$

b. Linear Discriminant Analysis

LDA approaches the problem by assuming that the conditional probability density functions $p(x|y = 0)$ and $p(x|y = 1)$ are both normally distributed with mean and covariance parameters. Under this assumption, the Bayes optimal solution is to predict points as being from the second class if the log of the likelihood ratios is bigger than threshold 0.5.

c. K-Nearest Neighbor

KNN identifies k users with features that are most similar to the target user measured by Euclidean distance.

d. Support Vector Machine

SVM constructs a hyper-plane or set of hyper-planes in a high dimensional space, which can be used for classification tasks. rbf kernel was used which is given as:

$$K(x,x') = \exp(-\frac{||x-x'||^2}{2\sigma^2})$$

e. Decision Trees

In Decision Trees, the input space X is repeatedly split into two child regions by thresholding a single feature. Given a parent region R_p, and two children region R_1 and R_2 resulting from a given split on a single feature, the decrease in loss is given by:

$$L(R_p) - \frac{|R_1|L(R_1) + |R_2|L(R_2)}{|R_1| + |R_2|}$$

f. Random Forests

Random Forest Classifiers are constructed by bagging Decision Tree Classifiers which are trained using a random subset of features.

g. Adaptive Boosting

AdaBoost classifiers sequentially apply base classifiers to modified versions of the data. The initial version of the data has uniform weighting. In each successive iteration m, observation weights are modified to place more weight on examples misclassified by the previous classifier $G_{m-1}(x)$. Decision Tree Classifiers are used as the base classifier in this project.

h. Gradient Boosting

Gradient boosting classifiers are the AdaBoosting method combined with weighted minimization, after which the classifiers and weighted inputs are recalculated. The objective of Gradient Boosting classifiers is to minimize the loss, or the difference between the actual class value of the training example and the predicted class value. It isn't required to understand the process for reducing the classifier's loss, but it operates similarly to gradient descent in a neural network.

i. Extra Trees Classifier

Extra Trees is like Random Forest, in that it builds multiple trees and splits nodes using random subsets of features, but with two key differences: it does not bootstrap observations (meaning it samples without replacement), and nodes are split on random splits, not best splits.

j. Multi-layer Perceptron

Multi-layer Perceptrons are fully connected neural networks characterized by a set of hidden layers, activation functions, and an optimizer used during back propagation.

Hyperparameter tuning of SVM, Random Forest, Adaboost, Gradient Boosting and Extra Trees classifiers was done over various parameters via grid search with criterion of validation set accuracy. The final models selected based on validation set performance were then evaluated on the test set.

VI RESULTS

10 fold cross validation was done on all models for evaluation. The best performing models were then evaluated on 30% split test set.

	Mean Validation Score	Testing Accuracy
Logistic Regression	0.683639	0.664365
LDA	0.681863	0.662983
KNN	0.608717	0.623619
SVM	0.660258	0.665055
Decision Tree	0.610786	0.591160
Random Forests	0.677445	**0.689917**
AdaBoost	0.608711	0.590470
Gradient Boosting	0.689871	0.680939
Extra Trees	**0.691063**	0.685773
MLP	0.674486	0.654696

Table 1 Comparison of models under evaluation

Upon hyperparameter tuning:

	Best Validation Score	Testing Accuracy
SVM	0.696976	0.675414
Random Forests	0.694323	0.685083
AdaBoost	0.608711	0.590470
Gradient Boosting	**0.706759**	**0.688536**
Extra Trees	0.692557	0.681630

Table 2 Comparison of tuned models

Extra Trees classifier achieved the highest mean cross validation accuracy of 0.691063, whereas Random Forests achieved maximum testing accuracy of 0.689917. Tuned Gradient Boosting classifier performed best with highest validation score of 0.706759 and testing accuracy of 0.688536

Figure 16 shows the most important features responsible for the tuned models. All models have the same feature as the most important.

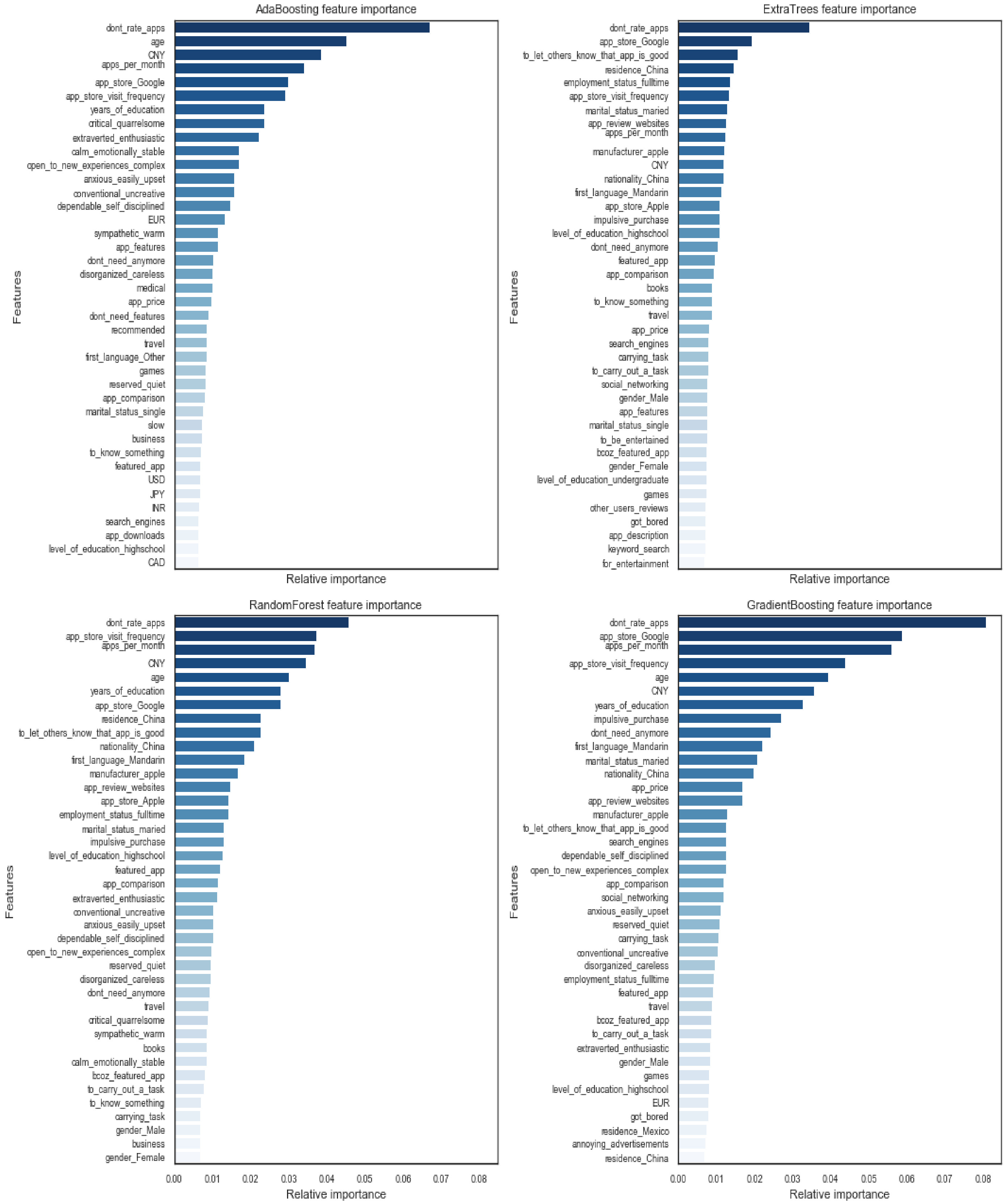

Figure 16 Relative importance of first 40 features of AdaBoosting, Random Forest, Extra Trees and Gradient Boosting classifiers. 'don't_rate_apps' is the most important feature common to all the four classifiers. Although there is some change in the ranking of the later features, those features appear in all of the classifiers.

VII MODEL INTERPRETABILITY

State-of-the-art model interpretability libraries SHAP, Eli5 and Permutation Importance were used to study the models and their predictions based on the important feature values. SHAP’s Tree Explainer was used on Extra Trees Classifier to interpret the output.

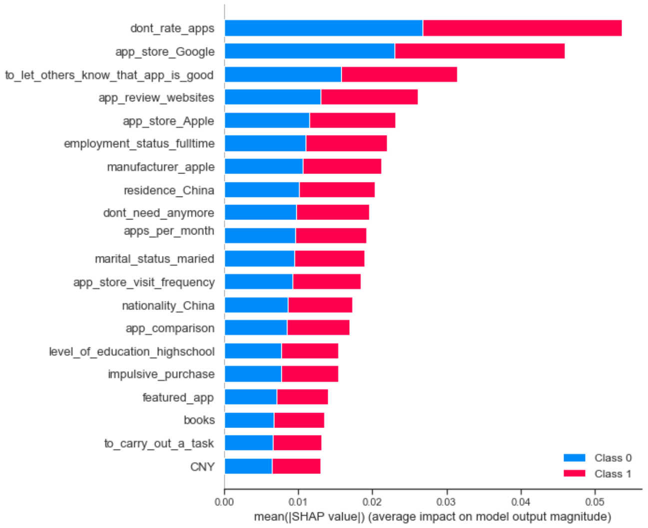

Figure 17 Summary plot with stacked bars showing binary class output and the mean SHAP values

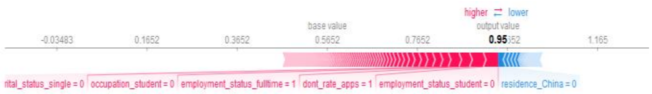

Figure 18 SHAP force plot with expected value 0.95

Figure 18 shows the output force plot for an observation with predicted expected value of 0.95. The force plot shows features each contributing to push the model output from the base value (the average model output over the training dataset we passed) to the model output. Features pushing the prediction higher are shown in red, those pushing the prediction lower are in blue. For the observation shown in figure 18, features that the user is not a student, doesn’t rate apps, is employed full-time, etc. worked in favor, whereas features that the user is not a resident of China, etc. worked against the expected outcome.

VIII CONCLUSION

In this project, binary classification models were created with the end goal of predicting payment behavior of mobile app users. Linear models, tree based models and neural networks were trained. Random Forest, Extra Trees and Gradient Boosting classifier were found to be the top performing models.

Recommendations can be made to businesses and software engineering teams on improvements they can make to better engage with the users.

IX FUTURE WORK

The models achieved good results. In order to improve model performance, users can be grouped by personalities or type of countries i.e. developed and developing, and modeled separately to create more accurate predictions and section the market for more actionable insights. The model output suggests that more rigorous feature engineering could improve accuracy and simultaneously reduce computational cost.

ACKNOWLEDGEMENT

Thank you to Dr. Kevin Ding for all the continuous support with this project.

CODE: github.com/jaysmitjadhav/Will-They-Pay