Cindy Wang

Jianqingwang\_cindy@yahoo.com

Abstract

Can the first 2 hours experience already tells us whether a user would bill through after 14-days free trial? Maybe the very first thing that we ask user to do already improve or decrease the chance?

EArly Indicator of Bill Through 14 days later

Data Science Final Project

**Executive Summary**

As a typical data science project, the study started with a well- defined business problem “What are early indicators for free trial billthrough 14 days later so that we can act or course correct early?” The study went through the full cycle of data understanding, data preparation, modeling and evaluation. Not surprisingly, as part of iterative process, learning from original project led to new problem (question) and a second data science project. In the end, the conclusion is that to improve the billthrough, Ancestry.com can change user behavior as early as when a user created his very first tree node by purposely collecting certain facts about the node or suggest the user starting building tree with certain type of nodes. This act would improve the chance of Ancestry hint generated for the user automatically and enable the user to get to his first meaningful/rewarding family discovery sooner.

**Starting Problem statement**

Ancestry.com is running subscription based model and acquiring new users by offering 14-days free trial. As the result of that, business always want to understand what factors driving bilthrough rate of free trials, especially early indicators. The earlier we can spot the possible future cancelers, more time we have to take actions accordingly. Factors falling into early indicators include how users became site registrants (pre-requirement for any site purchase), products they chose at signup and their first 2 hours site experience and engagement. This study focus on those factors.

**Dataset and Data processing**

Based on various past analysis, we know free trial billthrough were influenced by many other factors that were not necessarily the interest of this study, but potentially disguise the factors in this study if we don’t control them. As the result, specific cohorts of free trailers were chosen free from those influencers, such as major Marketing program/email and etc.

The data to build features/variables used in this study originated from multiple data systems. Most of data, such as user registration and product purchase data, have been moved into enterprise data warehouse over time. However, engagement data had been ETLed from multiple product feature databases. They resided in different tables and never integrated at user level. Especially focusing on first 2 hours required to sequence all events from multiple product feature database, which may not be synced among them and not with registration/purchase data. After finding the data sync issue and ironing out timestamp among different systems, 14Million post signup activities were aggregated at user level.

Feature engineering work focused on creating variables to represent different hypothesis to be tested. The consumers of the study is Product Managers who were product feature focused. The study not only need to find early indicator, but also identify what type of user engagements should be encouraged. For example, one hypothesis was that if a user was able to add new tree nodes by discovering them in content offered by Ancestry early on, he would be impressed and had more chance to bill through. This could be used potentially to justify prioritizing the content with family member discovery higher than those without. In the end, there were about 60 variables (See Appendix A).

Variables that have a lot of missing value, for example, Age, were removed from modeling exercise.

All the dataset manipulation and variable creation were handled in SQL. Once it was done, the final modeling dataset were loaded into csv and read into Python environment. As part of this process, all the missing value were saved as string value “(null)” for a given variable. So technically there were no missing values for all variables.

Categorical variables were converted to numerical variables using sklearn preprocessing.LabelEncoder(). For existing numerical variables, to improve model performance, the covariance among them were explored (after values were scaled). Only one of those variables was kept in the final dataset. For example, both nodbcitationcountin2hours and nodbcitationnodescountin2hours See covariance matrix table at Appendix B.

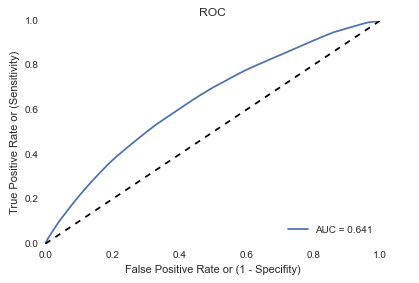
**Modeling**

Due to the fact that the priority of the study is to explain the result to human, not predict, Decision Tree was used. Since the overall billthrough rate was 41%, relative balanced dataset, the model just built on original dataset.

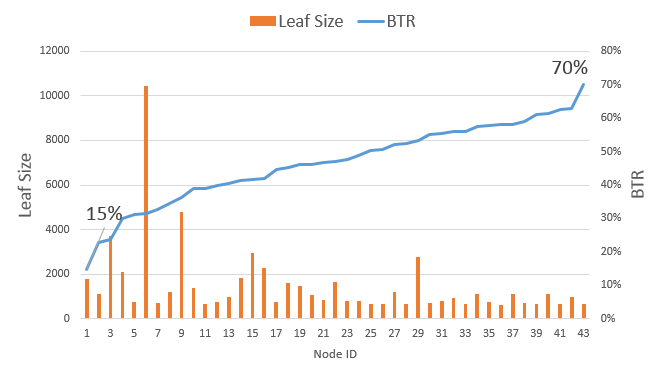
To avoid model over fitting, cross validation was used to select optimal value for sklearn.tree DecisionTreeClassifier’s parameter max\_depth and min\_samples\_leaf. 2~19 were tested for Max\_depth and 1%~19% were tested for min\_samples\_leaf. Average accuracy score was used to select the best parameter value. Optimal Max Tree Depth is 6 and min samples leaf: 1%.

The performance of model after fitting on the whole dataset is:

* Accuracy score: 62.3%
* Precision: 57%
* True Positive Rate (Recall): 38%
* False Positive Rate 20.1%
* AUC of ROC curve: 0.641



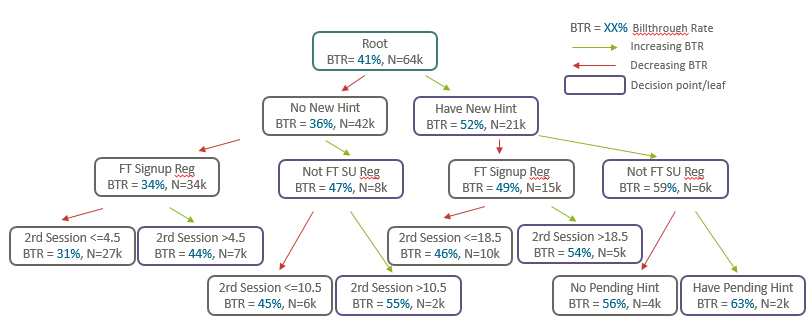
Even though AUC is not very high, the decision can still segment users into groups with very different billthrough rates. The best node has 70% billthrough rate while the worst node has 15% billthrough rate.



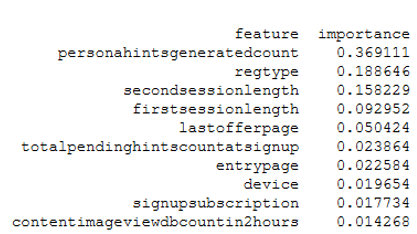
The most important leading indicator was whether there were new hints generated within the first 2 hours after free trial starts. The best leaf nodes actually have 7+ hints generated. For those who are not familiar with Ancestry product, having tree hints indicated that our engine can automatically find content from our content database or other member tree to help the given user to grow his tree, without he actively search for it.

The length of the first session and second session were other important engagement related variables. Other non-engagement related variables picked by model were registration type, offer page, entry page to site, device and package a user chose matters as well

Tree can be visualized with decision criteria, node size and related billthrough rate below.



See top 10 features with their importance below



**Further Investigation**

Now the question changed to “What triggered the hint generation? What are minimal facts that users need to provide for a tree node to get a hint generated?”

New question leads to research into how hint engine works and whether we have any data. The ideal dataset should include both successful and failed hint request and the node input from user that triggered the hint request. Luckily Ancestry has the service log, even though it is not readily available in the data warehouse.

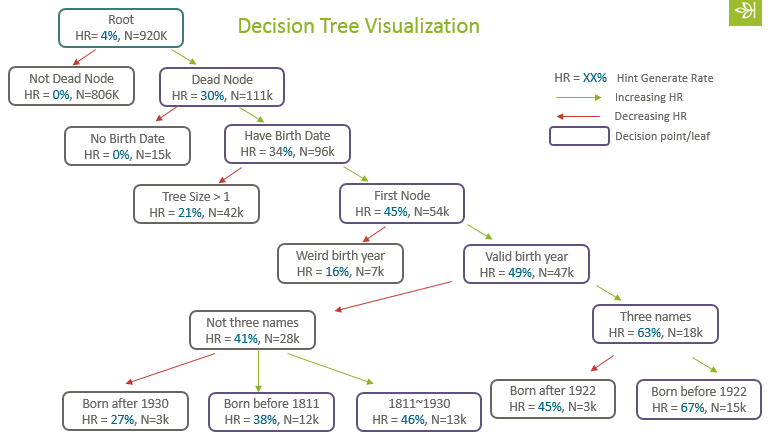
The features represents facts put into node by user at the time of hint request issued. They are Gender, # of Names (First, Middle and Last), # of relatives (# of other nodes in tree), # and type of facts (residence, military, marriage and etc.), birth date and location and death date and location.

The dataset includes 917K instance and 23 variables. There was no null value. Decision Tree model is used so output is understandable by human. The dataset was unbalanced with overall hint generation rate about 4%. Running model with class\_weight= 'auto' will adjust for unbalanced dataset. However, I also ran the model without class\_weight= 'auto'. In term of variables selected, the outcome was similar between two. However, class\_weight= 'auto' render a model with less leaf nodes, 99.7% recall (instead of 47.6%) with the sacrifice of precision 36% (instead of 62%).

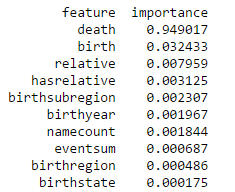
The performance of model after fitting on the whole dataset is:

* Accuracy score: 93.5%
* Precision: 36%
* True Positive Rate (Recall): 99.7%
* False Positive Rate 6.7%
* AUC of ROC curve: 0.984

It turned out that for hint to be generated, a node must have death date and birth date. Once these two conditions were satisfied, the chance of getting hint improved if the dead node was the first node, have three piece names (first, middle and last) and born before 1922.

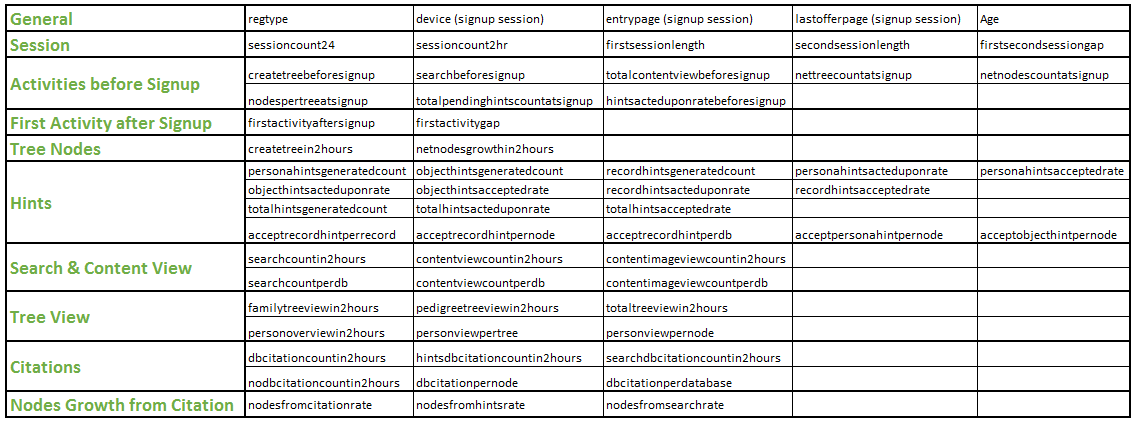


See top 10 features with their importance below



To sum up, there were early indicators of bill through 14 days later. They included registration type, offer page, entry page to site, device and package a user chose for his free trial. Once free trial started, the most important engagement indicator during first 2 hours was whether a hint was generated. The first session and second session were other important engagement related variables. Further investigation on the minimal node input to trigger hint generation revealed that the first node being diseased person with birth date before 1922 and three pieces of names had the best chance to get hint. This insight can be potentially used to change how Ancestry guide users building their tree, by suggesting them starting with diseased ancestor and getting all key facts before moving on to next tree node.

**Appendix A**



|  |  |  |
| --- | --- | --- |
| **Variables** | **Variable Group** | **Description** |
| regtype | Registration,  Signup, User data | Registration Type, for example, Merlin or FT signup |
| Age | Age is calculated based on tree owner's corresponding mepid in the tree. |
| entrypage | Omniture Data | Entry page of the signup session |
| device | Device of the signup session |
| lastofferpage | Last offer page before signup (90% match rate) |
| sessioncount24 | Sessions After Signup | # of sessions within 24 hours after signup |
| sessioncount2hr | # of sessions within 2 hours after signup |
| firstsessionlength | first session length In Minutes |
| secondsessionlength | second session length In Minutes |
| firstsecondsessiongap | gap between first session and second session in Minutes |
| createtreebeforesignup | Activities Before Signup | Whether tree was created before signup |
| searchvolumebeforesignup | # of searches before signup |
| contentviewimagebeforesignup | # of image content view before signup |
| contentviewimagedbbeforesignup | # of database being image content viewed before signup |
| contentviewtextbeforesignup | # of text content view before signup |
| contentviewtextdbbeforesignup | # of database being text content viewed before signup |
| contentviewhoverbeforesignup | # of hover content view before signup |
| contentviewhoverdbbeforesignup | # of database being hover content viewed before signup |
| contentviewhovernosubbeforesignup | # of hover-no-sub content view before signup |
| contentviewhovernosubdbbeforesignup | # of database being hover-no-sub content viewed before signup |
| hintsacteduponratebeforesignup | (all hints accepted + all hints rejected )/all hints generated before signup |
| grosstreecountatsignup | Snapshot at Signup | gross # of trees at the moment of signup |
| nettreecountatsignup | net # of trees at the moment of signup |
| netnodescountatsignup | net # of nodes at the moment of signup |
| totalpendinghintscountatsignup | # of pending hints at the moment of signup |
| firstactivityaftersignnup | Activities witthin 2 Active Hours | First activity after signup, for example, content view, search, create tree and etc. |
| createtreein2hours | # of tree created |
| netnodesgrowthin2hours | net # of nodes growth (K) |
| personahintsgeneratedcount | # of tree hints generated |
| personahintsacteduponrate | (tree hints accepted + tree hints rejected )/ tree hints generated |
| personahintsacceptedrate | tree hints accepted / (tree hints accepted + tree hints rejected ) |
| acceptpersonahintsnodescount | # of nodes tree hints accepted on |
| objecthintsgeneratedcount | # of object hints generated |
| objecthintsacteduponrate | (object hints accepted + object hints rejected )/object hints generated |
| objecthintsacceptedrate | object hints accepted / (object hints accepted + object hints rejected ) |
| acceptobjecthintsnodescount | # of nodes object hints accepted on |
| recordhintsgeneratedcount | # of record hints generated |
| recordhintsacteduponrate | (record hints accepted + record hints rejected )/record hints generated |
| recordhintsacceptedrate | record hints accepted / (record hints accepted + record hints rejected ) |
| acceptrecordhintsnodescount | # of nodes record hints accepted on |
| AcceptRecordHintsRecordCount | Number of distinct records hints accepted on |
| AcceptRecordHintsDBIDCount | Number of distinct DBIDs hints accepted on |
| totalhintsgeneratedcount | # of all hints generated |
| totalhintsacteduponrate | (all hints accepted + all hints rejected )/all hints generated |
| totalhintsacceptedrate | all hints accepted / (all hints accepted + all hints rejected ) |
| acceptrecordhintperrecord | # of record hints accepted/ # of records used by those record hints |
| acceptrecordhintperdb | # of record hints accepted/ # of databases used by those record hints |
| acceptrecordhintpernode | # of record hints accepted/ # of nodes record hints accepted on |
| acceptpersonahintpernode | # of tree hints accepted/ # of nodes tree hints accepted on |
| acceptobjecthintpernode | # of object hints accepted/ # of nodes object hints accepted on |
| searchcountin2hours | # of searches |
| searchdbcountin2hours | # of database searched |
| contentviewcountin2hours | # of content view |
| contentviewdbcountin2hours | # of database being content viewed |
| contentimageviewcountin2hours | # of image content view |
| contentimageviewdbcountin2hours | # of database being image content viewed |
| familytreeviewin2hours | # of family tree views |
| pedigreetreeviewin2hours | # of pedigree tree views |
| totaltreeviewin2hours | total # of tree views |
| personoverviewin2hours | # of person overviews |
| personoverviewtreesin2hours | # of trees having person overviews |
| personoverviewnodesin2hours | # of nodes having person overviews |
| dbcitationcountin2hours | # of citations with DBID created (A) |
| hintsdbcitationcountin2hours | # of citations with DBID attributed to hints |
| searchdbcitationcountin2hours | # of citations with DBID attributed to searches |
| dbcitationnodescountin2hours | # of nodes which citations with DBID were attached to (B) |
| dbcitationdatabasecountin2hours | # of database which citations with DBID were from (M) |
| nodbcitationcountin2hours | # of citations without DBID created |
| nodbcitationnodescountin2hours | # of nodes which citations without DBID were attached to |
| dbcitationpernode | A/B |
| dbcitationperdatabase | A/M |
| createnodesfromhintsin2hours | # of nodes created from hints citations (D) |
| nodesfromhintsrate | F/K |
| createnodesfromsearchin2hours | # of nodes created from search citations (F) |
| nodesfromsearchrate | G/K |
| createnodesfromotherin2hours | # of nodes created from other citations (G) |
| nodesfromcitation | D+F+G |
| nodesfromcitationrate | D/K |

**Appendix B**

