Text Analytics on Google Apps

Class: Text Analytics

Professor: Yilu Zhou

Team Members: Nan Wang, Anshu Vyas, Ellin Iqra, Chang Yin

Table of Contents

Executive Summary

This study is based on text analysis and data mining on text description and information about 2227 Google apps. We are wanted to label updated information for every app, accurately classify apps into ten categories and find out common attributes of apps with large installation number and high customer rating.

We first utilized Scrapy, an open source framework for extracting data from websites, to crawl data needed and then with the help of Python, we generated features from pure text to build predictive models (K-Means and Gaussian Naive Bayes). After that, we conducted prescriptive analysis, through data visualization skills fulfilled in Tableau to find out patterns that make apps successful.

To conclude, we are able to generate five comprehensive and accurate labels (Add New Features, GamePlay Modification, Improve Levels, Fix Bugs & New versions, Fix Minor Bugs & Optimization) for app updated information and classify apps with relatively high accuracy (72%). Moreover, it is observed that apps falling into the subcategory of Strategy, developed by top developers and operated on flexible systems are most likely to be successful. These apps also generally have high installation number of over 50 thousand and review ratings of over 4.0.

Business Goal Analysis

Mobile applications have turned into an enormously profitable business, with revenue from mobile applications expected to exceed fifty billion USD by 2016[1]. These profits are not distributed equally amongst developers, with forty-seven percent of developers making less than one-hundred USD, more than half of which make nothing at all [2]; creating a successful application is not easy. Moreover, with the wide explosion of apps and their high upgrade speed, it is necessary and of commercial interest to classify them into specific categories and to label their update information, for convenience and quickly search and customer availability. Luckily, an enormous amount of data on mobile applications is made available by Google on its app stores. We can gather this data and use text mining techniques to offer solutions to problems addressed.

Text mining techniques have been applied to many fields nowadays. One of the application is, by using text mining, to propose a decision support system to aid movie investment decisions at the early stage of movie productions. In this study, text analysis is applied to get information from app descriptions online in order to gain investment-aided insights of what kinds of apps are more likely to be successful.

Moreover, this project is aimed at categorizing Android apps based only on their description texts. There are over twenty subcategories of Android apps. We hope to help categorize apps with higher accuracy and efficiency. In addition, it is noticed that apps are updated with high frequency. We want to catch and summarize every update so that app users would, just by a quick look, form an overview of what their apps are updating about.

We believe that with sufficient data crawling from the website and application of both text mining and data mining techniques, we will be able to help improve app-users experience and support decision process for app investment.

Dataset Description

Data Crawling

Google makes data available about its applications on Source: https://play.google.com/store/apps/category/GAME/collection/topselling_new_free. These pages contain data which can be extracted such as the name of the application, the description, the number of installations, the average rating of the application, and many more features.

For each desired feature, a feature extracting function was written using Scrapy, which, given an input HTML file, would output just the feature in question. For all feature extracting functions and all output HTML files, features were crawled and the resultant dataset is stored in the format of csv.

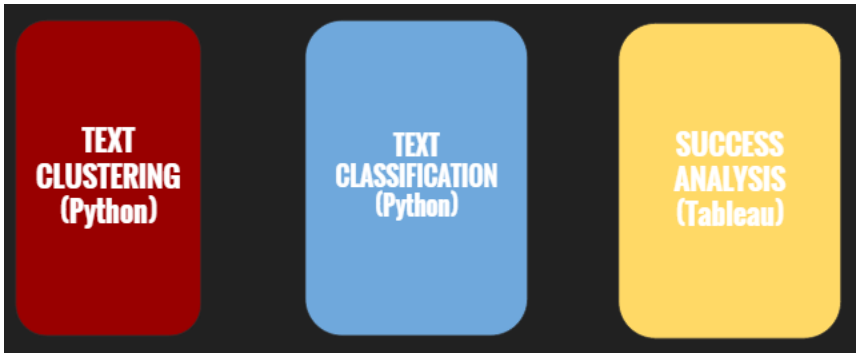
The whole dataset is about app information containing 2227 google apps from March 1 to April 21.

Feature Extraction

Features extracted include publish date, subcategory, software version, operating systems, average user rating, number of {5, 4, 3, 2, 1} star ratings, number of installations, description, name of the application, whether or not the developer is a top developer, and the size of the Android Application Package (APK).

System Design

Overall, our project includes three phrases. One is to transform text and generate features through applying text mining techniques, one is to build two models (K-means clustering and Gaussian Naive Bayes classification) through using data mining skills and the most phase is applying data visualization techniques to do prescriptive analysis on app success.



Text Preparation

In this process, we clean the dataset, read text into data-frame with the help of python Pandas package and then remove numbers as well as customized stopwords, lowercase characters, do tokenization and word stemming with the help of NLTK package

```
%timeit
def remove_numbers(s):
    return s.translate(None, string.digits)

def lowercase_remove_punctuation(s):
    s = s.lower()
    s = s.translate(None, string.punctuation)
    return s

def remove_stopwords(s):
    token_list = word_tokenize(s)
    exclude_stopwords = lambda token : token not in stopwords
    return ' '.join(filter(exclude_stopwords, token_list))

def stem_token_list(token_list):
    STEMMER = PorterStemmer()
    return [STEMMER.stem(tok.decode('ascii',errors = 'ignore')) for tok in token_list]

from nltk import pos_tag
def tag(token_list):
    POS = pos_tag(token_list)
    return [word for (word, tag) in POS if tag.startswith('N')]

def restring_tokens(token_list):
    return ' '.join(token_list)

def all_work(s):
    s = remove_numbers(s)
    s = lowercase_remove_punctuation(s)
    s = remove_stopwords(s)
    token_list = word_tokenize(s)
    #token_list = tag(token_list)
    token_list = stem_token_list(token_list)
    return restring_tokens(token_list)

def all_work_for_whatisnew(s):
    s = remove_numbers(s)
    s = lowercase_remove_punctuation(s)
    s = remove_stopwords(s)
    token_list = word_tokenize(s)
    token_list = tag(token_list)
    token_list = stem_token_list(token_list)
    return restring_tokens(token_list)
```

Figure 1: Python Code Example of Text Preparation

```
 u'play one success la vega slot game comfort home go mobil devic impress princess win wealth
glori golden knight scatter buck stack wild give quest rich reward download free collect welc
om bonu get start',
 u'kitti pawp avail play android',
 u'nan',
 u' new everi tile get bjoker tileb use wise onlin leaderboard googl play game bugfixesani su
ggest bug report welcomepleas write bad review contact us email ijdpappsgmailcomi',
 u'                ',
 u'new version alway work hard make video poker delux best video poker game fix screen flicke
r happen devic fix number bug first releas super stablekeep send feedback keep make game bett
er thank play best video poker game around',
 u'bugfix stabil improv',
 u'lot great improv updat ad brand new enemi upgrad graphic particl effect improv music audio
better score feedback anim time color ad support older android versionsbug fix fix case game
slow fix issu could caus camera shake effect continu heavi attack',
 u'level bug fix',
 u'remov slot extrem ad button restartthi cours everyth continu updat improv game',
 u'decreas ad frequenc',
 u'amaz gameplay',
 u' ',
 u'releas beta live close beta commun thank supportst fix mani crash freez bug fixedgam play
spark attack effect halv new arena backgroundsnew extend tutori select how play earn reward l
earn advanc concept game',
 u'graphic engin updatedfix imag doubl oneplu devic graphic artifact work gyroscop',
 u'first thank much play lost harmoni messag kind reviewson new version support nexu bug fix
tweak improv',
 u'ad levelsadjust difficulti levelsbug fixesremov ad',
 u'perform enhanc critic bug fixesthank play droppi ball',
 u'thank love continu work game adjust reviv mechan',
 u'fix certain bug',
```

Figure 2: Sample of Prepared Text

Model Building

Model Building Part 1 - Clustering to label app update information

1. Feature Engineering

    1.1. TFIDF, N-gram and Minimum Document: In this step, we are try to extract features from text. Based on back and forth trials and careful observation of text, we decide that minimum documents of four (small overlap of words usage in different app update text) and N_gram in the range of one to five (many words appear together with high frequency) would be the best for our model performance. 135 features are generated.

```
# vectorize and re-weight
desc_vect = tfidf_vectorizer.fit_transform(what_is_new)
coo = desc_vect.tocoo(copy = False)
df_word_matrix = pd.DataFrame({'index': coo.row, 'feature': coo.col, 'data': coo.data}
                 )[['index', 'feature', 'data']].sort_values(['index', 'feature']).reset_index(dr
word_matrix = df_word_matrix.pivot('index', 'feature', 'data')
word_matrix = word_matrix.fillna(0)
```

```
word_matrix.head()
```

feature	0	1	2	3	4	5	6	7	8	9	...	125	126	127	128	129	130	131	132	133	134
index																					
1	0	0	0	0	0	0.000000	0	0	0.000000	0	...	0	0	0	0	0.000000	0.000000	0	0.290277	0	0.000000
2	0	0	0	0	0	0.652527	0	0	0.627365	0	...	0	0	0	0	0.000000	0.000000	0	0.000000	0	0.000000
4	0	0	0	0	0	0.000000	0	0	0.000000	0	...	0	0	0	0	0.000000	0.000000	0	0.000000	0	0.000000
6	0	0	0	0	0	0.000000	0	0	0.000000	0	...	0	0	0	0	0.129226	0.493974	0	0.000000	0	0.164658
7	0	0	0	0	0	0.000000	0	0	0.000000	0	...	0	0	0	0	0.000000	0.000000	0	0.000000	0	0.000000

5 rows × 135 columns

Figure 2: Python Code for Generating Features

```
[(u'minor bug fix', 1.0),
 (u'us', 0.92468964501509099),
 (u'player', 0.80887092392482673),
 (u'multipl', 0.77569328502893675),
 (u'spin', 0.76812347947729964),
 (u'updat new', 0.74707184593519893),
 (u'find', 0.67876051148278016),
 (u'updat', 0.66474330159165551),
 (u'bonu', 0.65252742204348113),
 (u'sound', 0.64030174158414288),
 (u'stabil', 0.63363569815498977),
 (u'mission', 0.63111007562945531),
 (u'bug', 0.62736513460690246),
 (u'first', 0.6236336038293504),
 (u'user experi', 0.60771003503531418),
 (u'increas', 0.60470548038951843),
 (u'user', 0.58710214295216057),
 (u'includ', 0.5598065947012455),
 (u'store', 0.55586785683546935),
 (u'thank', 0.51545319048405958)]
```

Figure 3: Top 20 TFIDF Terms

1.2. Part of Speech: In order to select effective features and get rid of redundant features, we applied the technique of part of speech (a category to which a word is assigned in accordance with its syntactic functions.) English words like I, they, am, is, are, those with no actual meanings, have been removed before building the machine learning model.

CD	1
FW	1
IN	2
JJ	26
JJR	1
NN	75
NNS	3
PRP	1
RB	2
RBR	1
RBS	1
VB	6
VBD	1
VBN	2
VBP	12

Figure 4: Distribution of Different POS

Cardinal number(CD), Determiner(DT), Foreign word(FW) have been removed.

1.3. Principal Component Analysis - Principal component analysis was performed on the inputs to the K-means models. According to the chart below, we can conclude that ninety principal components can explain all the variance in all features.

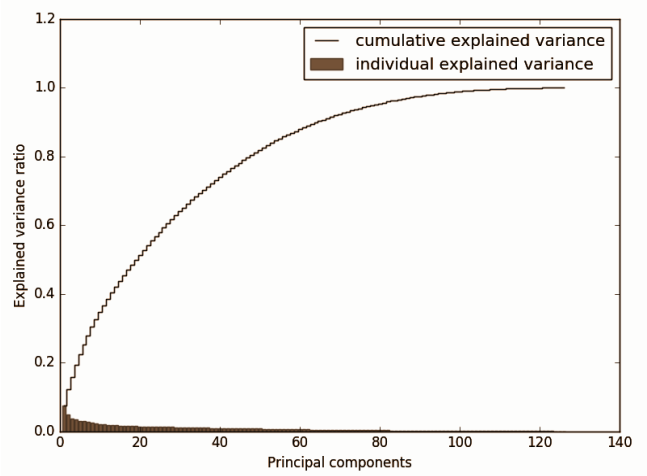


Figure 5: Cumulative Explained Variance

2. Machine Learning Model

k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. *k*-means clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

For update information clustering, we apply the algorithm of K-Means in this project.

```
sklearn_pca = PCA(n_components=90)
word_matrix_pca = sklearn_pca.fit_transform(word_matrix_less_features)
```

```
from sklearn.cluster import KMeans
km = KMeans(n_clusters = 4)
%time
km.fit(word_matrix_pca)
```

Figure 6: KMeans sample python codes (Tableau)

Model Building Part 2 - Classification to Categorize Apps

1. Category Choosing

There are in total 23 categories of apps in the dataset. To ensure the accuracy of classification model, we decide to just choose top ten popular categories of apps,

3. Feature Engineering

3.1. TFIDF, N_gram and minimum document: It is observed that for app description text, different word usage is preferred. Moreover, words tend to not appear together with high frequency. Thus to ensure the enough number of features, we set the minimum document number being 2 and the N_gram range being from 0 to 2 when generating features.

```
tfidf_vectorizer = TfidfVectorizer(
    min_df= 2,  # min count for relevant vocabulary
    max_features=100000,  # maximum number of features
    strip_accents='unicode',  # replace all accented unicode char
    # by their corresponding  ASCII char
    analyzer='word',  # features made of words
    token_pattern=u'[a-z]+',  # tokenize only words of 4+ chars
    ngram_range=(1, 2),  # features made of a single tokens
    use_idf=True,  # enable inverse-document-frequency reweighting
    smooth_idf=True,  # prevents zero division for unseen words
    sublinear_tf=False)
```

Figure 7: Features for Classification

3.2. Part of Speech: In order to select effective features and get rid of redundant features, we apply the technique of part of speech (a category to which a word is assigned in accordance with its syntactic functions.) English words like I, they, am, is, are, those with no actual meanings, have been removed before building the machine learning model.

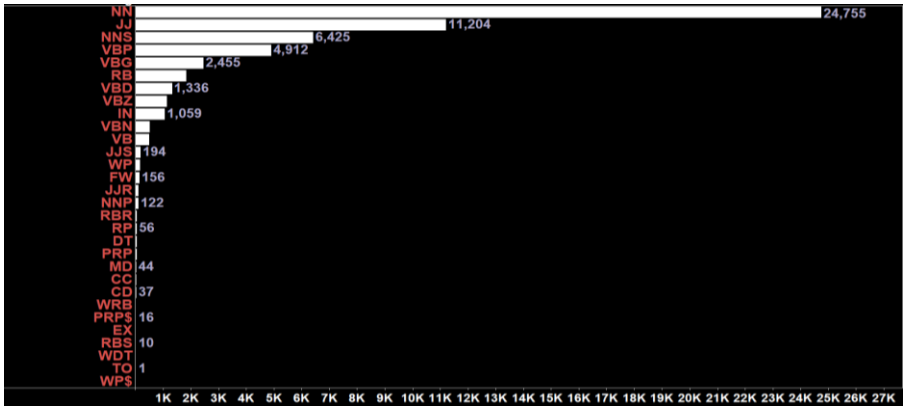


Figure 8: Part of Speech

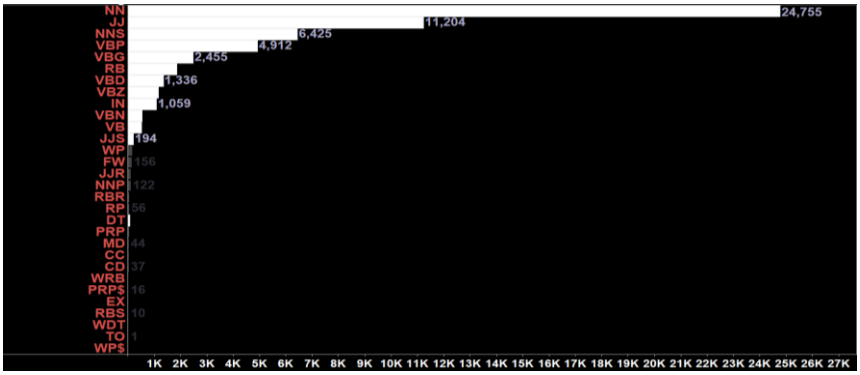


Figure 9: Selected Part of Speech

Words with part of speech, WP, FW, JJR, NNP, RP, PRP, MD, CC,CD, WRB, PRP$, EX, RBS, WDT, TO AND WP$ have been removed before building the classification model.

3.3. TFIDF Distribution: After checking the TFIDF distribution, we decide to remove features with TFIDF lower than 0.074 (mean) and higher than 0.3.

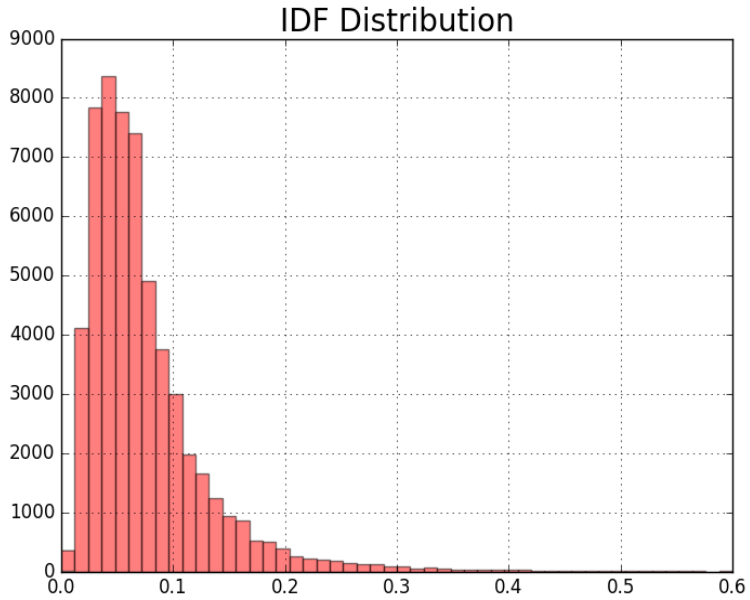


Figure 10: TFIDF Distribution

At this point, feature number has been reduced from 51000 to 20040.

4. Machine Learning Model

We use Gaussian Naive Bayes to build the classification model.

GaussianNB implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Prescriptive Analysis

During this process, we apply data visualization skills to find out correlation between app success and app subcategory, whether or not developed by top developers as well as operating system, so that it can be concluded that which type of app has higher possibility of being successful.

System implementation

Dataset Collection

The Data we want to collect is in the format is as figure 11.

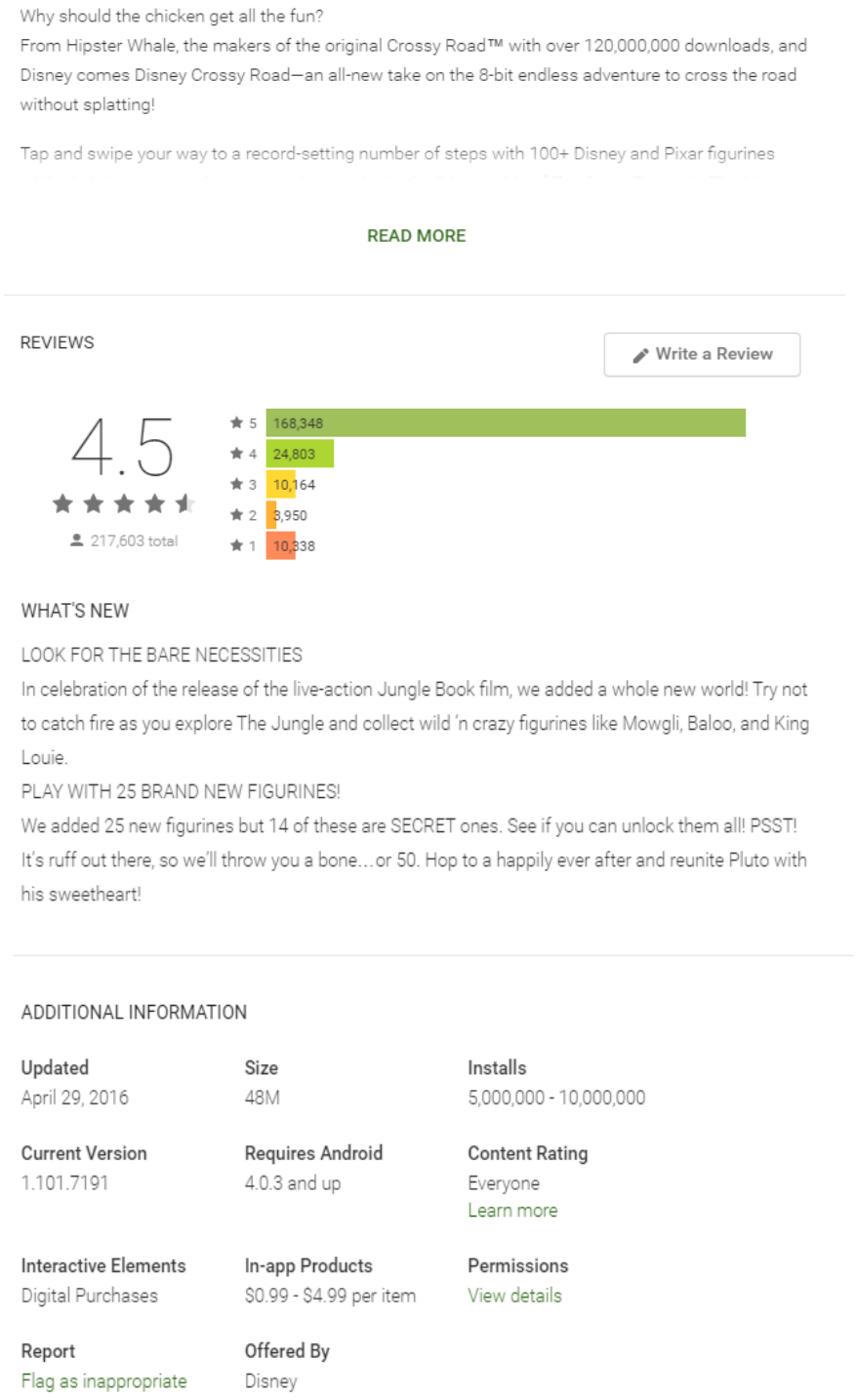


Figure 11: Web Format Data Sample

By using Scrapy, we crawl all the information needed and save it in the format of csv as the following screenshot shows.

```
operatingSys,subcategory,description,what_is_new,url,NumInstall,DatePublished,softwareVersion,size,topDeveloper,reviews_num,Num_star,app_na
      4.0.3 and up    ,Casual,"If you work,you lose! ,So maintain your kept man life and get lots of allowance!,-FAQ,Q.DPS in background is
      4.0.3 and up    ,Strategy,"Enter the Arena! From the creators of Clash of Clans comes a real-time multiplayer game starring the Royal
      3.0 and up    ,Pretend Play,"Welcome to the cutest store in town!,Kiki, our little panda, his mom and some of his friends are waiting
      4.0.3 and up    ,Creativity,"Budge Studios™ presents Miss Hollywood: Vacation! After winning a dream getaway to visit a beautiful isl
      4.0.3 and up    ,Role Playing,"★★★ Medal Masters ★★★,■ Pick Up and Play RealTime Casual RPG,  • Experience indepth RPG action with
      2.3 and up    ,Simulation," Want a mobile game which has all vehicles in it? Look no further! ,Driving School 3D Highway Road has all
      2.3 and up    ,Strategy,"Like the song says, good guys finish last, and in this case, they also finish homeless and on fire. ,Welcome
      4.0.3 and up    ,Simulation,"Welcome to the sweetest city on the Google Play store! Build your very own candy world with countless op
      2.3 and up    ,Card,"This solitaire app has Klondike Solitaire 1 card and Klondike Solitaire 3 card draw. ,To make the game much cool
      4.0.3 and up    ,Word,"Monkey Wrench is a fun word find with a twist - you have to use the clues to figure out what the hidden words
      4.2 and up    ,Action,"UNIQUE SYMBIOSIS OF MUSIC AND GRAPHICS,Dub Dash is a fast-paced rhythm based action game. The tracks are decom
      3.0 and up    ,Puzzle,"Cookie Crush Mania is awesome sweet game with lots of fun ,Enjoy interesting and fun gameplay with lots of lev
      2.3 and up    ,Simulation,"Tractor Farming Simulator is the farm simulator where you can manage your own farm and harvest your crops.
      2.3 and up    ,Casual,"Are you a passionate about makeup, fix and dress your friends? If you consider yourself a lover of the overall
      2.3.3 and up    ,Role Playing,"King Online là một tuyệt phẩm được đánh giá là MMORPG hot nhất  2016 do CMN Mobile phát hành. ,Lấy bối
      2.3 and up    ,Role Playing,"Looking for a cool jumping game that has a fun gameplay and great mechanics?,Jump Idol, is here to offer
      4.0.3 and up    ,Role Playing,"NEVER SEEN ON MOBILE,The rebirth of the 3D first-person dungeon-crawler,Lead your team of mighty heroe
      3.0 and up    ,Role Playing,"• Get hooked on the gameplay from the very first minute.,Be amazed by the dazzling visuals and action at
```

```
df = df.reset_index(range(len(df)))
df.head()
```

	index	operatingSys	subcategory	description	what_is_new	NumInstall	DatePublished	softwareVersion	size	topD
0	0	3.0	Arcade	Stack up the blocks as high as you can!	nan	500000 - 1000000	2016-02-17	1	25	1
1	1	4.0	Role Playing	Juggernaut Wars is a new captivating Action RP...	- Much requested Exit window added,- Fixed a b...	10000 - 50000	2016-02-26	NaN	NaN	1
2	2	2.3	Casino	Mountains of Sichuan is a place where all thin...	nan	100 - 500	2016-02-23	NaN	NaN	0
3	3	2.3	Action	The beasts are back in Mutant Fighting Cup 2! ...	- Fixed an issue with the Multiplayer PvP syst...	100000 - 500000	2016-02-23	1.0.8	44	1
4	4	2.3	Adventure	Apple Shooter Champ is an exciting Shooting Ga...	nan	50000 - 100000	2016-02-05	1	12	0

Figure 12: CSV Format Data Sample

Here some exploratory analysis about the dataset before text mining and data mining process.

```
operatingSys:
19 unique values, 0 missing values
subcategory:
23 unique values, 0 missing values
description:
2814 unique values, 0 missing values
what_is_new:
224 unique values, 0 missing values
NumInstall:
16 unique values, 0 missing values
DatePublished:
91 unique values, 0 missing values
softwareVersion:
425 unique values, 576 missing values
size:
258 unique values, 599 missing values
topDeveloper:
2 unique values, 0 missing values
reviews_num:
2734 unique values, 0 missing values
Num_star:
33 unique values, 0 missing values
app_name:
2227 unique values, 0 missing values
DaysPublish:
91 unique values, 0 missing values
```

Figure 13: Exploratory Analysis

Text Preparation

As it is mentioned, we remove numbers as well as customized stopwords, lowercase characters, do tokenization and word stemming after reading all the data into Python Data Frame format with the help of pandas package. Sample text is shown below.

```
[u'add googl analyticsadd new level first world',
 u'minor tweak bug fixespleas rate game like tell friendspleas use ingam support featur report issu provid
feedbackthank playingpleas rate game like tell friendspl',
 u'releas beta live close beta commun thank supportst fix mani crash freez bug fixedgam play spark attack
effect halv new arena backgroundsnew extend tutori select how play earn reward learn advanc concept game',
 u'new version alway work hard make video poker delux best video poker game faster video poker play fix nu
mber bug first releasekeep send feedback keep make game better thank play',
 u'extra hour log game collect daili login bonu increas odd get pitcher train card pack fix issu player st
at temporarili unavail train upgrad player',
 u'nan',
 u'puzzl load speed optim fix ui tablet',
 u'small fix improv',
 u'level bug fix',
 u' new muscl car ad keep follow new updat new car level come',
 u'ui updat',
 u'hey escalatorsthank feedback reviewsthi updat includ epic charact collis fix screen changeskeep escal',
 u'optim',
 u'small fix',
 u'ad googl play leaderboardsad social media linksad nontim practic mode',
 u'name',
 u'initi releas',
 u'v updat prepar academi excit ingam event get readi epic battl releas new level level unlock amaz antman
mariah hill mani other sell unwant decor make room campu updat game balanc order appear mission board tune
pym particl better experi fix mani bug includ low memori crash video crash improv perform across board',
 u'version fix bug could auto save minor improv',
 u'version mar lot perform enhanc keep game run bettervers feb degre difficulti part level adjust minor bu
g fix perform optim',
 u'emili want play free demo version use demo play small part game test full game emili want play work dev
ic',
 u'enjoy multipl cat bug human around style pussi cat play smash object vibrant graphic realist hous envir
on indoor outdoor locat perform routin cat task chase annoy anim',
 u'new drift carsbonu gamestrain',
 u'hey guy great news app run smoother ever keep send us feedback keep creat best game join sunstorm famil
i httpwwwfacebookcomtabtal',
 u'tune bugfix',
 u'crossword avail also french german dutch new clue clue correct bugfix',
 u' new everi tile get bjoker tileb use wise onlin leaderboard googl play game bugfixesani suggest bug rep
ort welcomepleas write bad review contact us email ijdpappsgmailcomi',
 u'minor bugfix',
```

Figure 14: Prepared Text

Clustering for Update Information

Five clusters have been shown the best for clustering model performance, because numbers of records falling into each cluster are much similar (as shown in the figure 15) and each cluster is relatively unique and different from each other (as shown in the figure 16-20).

	what_is_new
cluster	
0	18
1	87
2	26
3	24
4	42

Figure 15: Five Clusters

Figure 16: Cluster 0

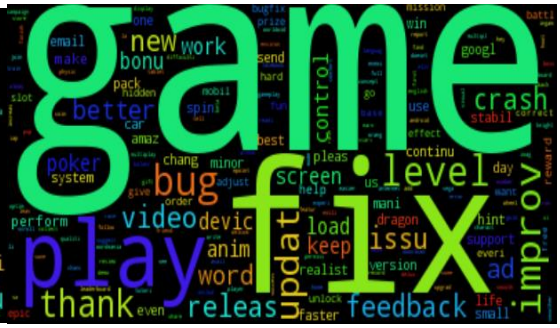


Figure 17: Cluster 1

Figure 18: Cluster 2

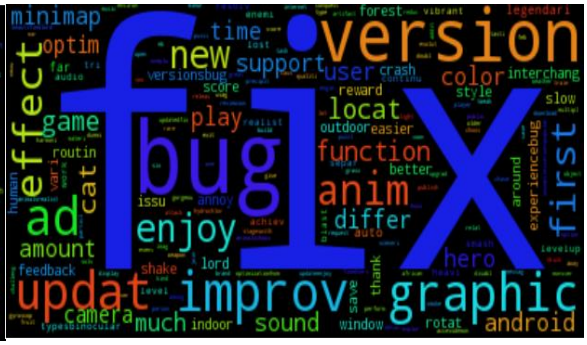


Figure 19: Cluster 3

Figure 20: Cluster 4

Classification for Categorizing Apps

With cross validation technique applied, 72% of accuracy has been achieved on the training dataset.

```
clf = GaussianNB()
#clf.fit(desc_matrix_less_features,description_for_text['subcategory'])
```

```
%timeit
scores_gb = cross_val_score(clf,
                            desc_matrix_less_features,description_for_text['subcategory'],c
                            v =10, scoring = 'accuracy')
```

```
scores_gb.mean()
```

```
0.71821866138887236
```

Evaluation

Clustering for Update Information

In the first cluster(figure 16), words as ‘add’,’new’,’menu’,’improve’,’option’ have stood out.

In the second cluster(figure 17),, terms like ‘game’,’fix’,’play’,’level’,’improve’,’release’,’crash’ have experienced high frequency.

In the third cluster(figure 18),, ‘new’,’level’,’card game’,’update’,’theme’,’feature’ and ‘multiplay’ have witnessed frequent occurrence.

In the fourth cluster(figure 19), expressions like ‘fix’, ‘version’,’graphic’,’effect’ are the most likely to be seen.

In the fifth cluster(figure 20),, ‘fix’,’minor’,’experience’,’enhance’,’interface’ and ‘performance’ have shown high frequency.

To conclude, clusters we generated as a whole differ from each other greatly and from frequent words, it can be easily deduced what each cluster is about. For example, we can say that cluster 0 is about adding new options or features into the gameplay apps while cluster 2 is about updating or improving apps into a new level.

However, the five clusters we end up with have some limitations. To begin with, there still exists obvious overlap between two clusters. For example, cluster 3 and cluster 4 are both about bug fixing. The small difference probably lies in cluster 3 focusing on fixing bugs to improve version (‘version’ is frequently shown.) while cluster 4 emphasizing on minor bugs to optimize apps (as ‘minor’ and ‘optimize’ appear a lot.)

Classification for Categorizing Apps

The accuracy of over 70% is not bad for classifying apps into 10 categories. However, though cross validation has been applied into the model building process to reduce the effect of overfitting, it is still expected to have overfitting problems for that app description intuitively contain different words and it is thus hard to summarize a common pattern for apps even of the same category.

App Success Analysis

As shown in figure 21, apps with the operation system of ‘varis with device’ are more likely to be successful, as they receive large number of reviews and positive reviews.

However, for apps with the operating systems of 3.2, though they have good customer reviews, the review number is not good enough, which means they are usually not widely-known despite offering good customer experience.

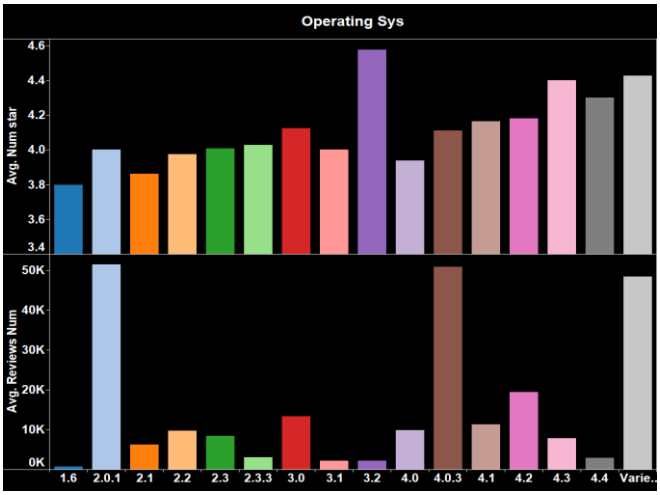


Figure 21: Review Num and Rating Star per Operating System (The data is filtered on number of installation over 50,000)

It can be concluded from the figure 22 that apps in the category of strategy enjoy the most likelihood of being successful.

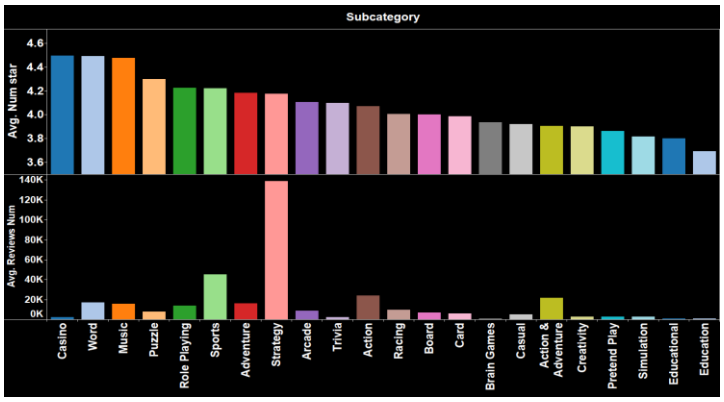


Figure 22: Review Num and Rating Star per Subcategory (The data is filtered on number of installation over 50,000)

We can see from figure 23 that customer experience on apps developed by top developers is slightly better than those not by top developers. However, top developers' products always receive high volume of reviews, which means they are usually eye-catching and widely-known. Therefore, apps developed by top developers are more likely to be noticed and thus successful.\

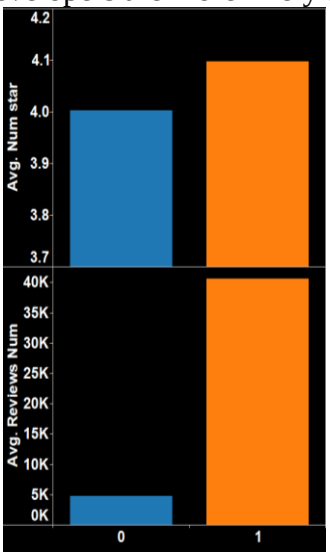


Figure 23: Review Num and Rating Star on Developer Type(The data is filtered on number of installation over 50,000)

Conclusion and Future Direction

Conclusion

We are able to draw these following conclusions based on our comprehensive analysis:

1. Five labels are available to cover most app update information
    a. Add New Features
    b. GamePlay Modification
    c. Improve Levels
    d. Fix Bugs & New Versions
    e. Fix Minor Bugs & Optimization

    With this finding, app updates can be labeled and customers would know quickly know that what this update is about rather than going through long texts and decide whether to update the app right now, for example, if the app is updated for minor bugs, it would be no interest of customers to waste time on updating it. But if a game app is adding in interesting features, customers would probably go and update the app to check new play experience.

2. Based on pure app description text, we can achieve over 70% of accuracy to classify apps into ten categories.
3. App Subcategory, Developer, Operating System have correlation with app success. If an app is in the category of strategy, with operating system of 'varies with device' and developed by top developer, this app is more likely to receive high number of positive customer reviews, which means it is more likely to be successful.

Future Direction

1. Broader Dataset Coverage

In this project, we only crawled apps in the google app store. In the future, we may touch upon Apps in the Apple store. Those two types of apps target with different customer groups. With a more diverse dataset collected, the result would be more valuable and interesting to see.

2. Lexical analysis for update information

Although clustering analysis enables us to find common patterns for app update information = automatically without any pre-defined words, we are still hoping to utilize some predefined keywords based on sense to help us to catch clusters more accurately and comprehensively. Accordingly, lexical analysis will be the possible solution with pre-defined category and keywords in this case.

3. Improved Model Accuracy for Classification

70% is far from satisfactory for a classification problem. In this future, with the help of customized dictionary and lexical analysis, a higher classification accuracy is expected.

References

[1] Worldwide mobile app revenues from 2011 to 2017 (in billion U.S. dollars). Statista. 2014. Web. 30 Nov. 2014. http://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/
[2] Maskey, Sameer. MapReduce for Statistical NLP/Machine Learning. 2012.
[3] Scrapy. http://doc.scrapy.org/en/latest/intro/overview.html 2014. Web. 01 Nov 2014.
[4] Selectors. Mozilla Developer Network Sept 2014. Web. 01 Nov 2014.
[5] Wilcox, Mark. Voskoglou, Christina. State of the Developer Nation Q3 2014. Vision Mobile. July 2014.