Data Mining for Business (BUDT758T)

Spam detection of Android apps post release on the App Store

Team Members:

Akhil Gupta

Harshit Gautam

Arpit Kapoor

Shreyansh Singhvi

ORIGINAL WORK STATEMENT

We the undersigned certify that the actual composition of this proposal was done by us and is original work.

	Typed Name	**Signature**
Contact Author	Akhil Gupta	
	Harshit Gautam	
	Arpit Kapoor	
	Shreyansh Singhvi	

Contents

Executive Summary

With more than 2 million apps on the Android store, Android marketplace is one of the highest revenue generating stream for Google. Approximately, thousands of applications are introduced in the market every day. With this frequency, the threat of spam applications being introduced in the market also increases. This is where our data analysis comes into picture to quarantine the spam apps and help Google and android users to continue having a seamless user experience.

Our project will help in identifying the spam applications present already in the market-place resulting in detection of such apps and then deleting them from the app-store.

Spam app detection and removal is of high priority for Google as bad user experience on the app-store may lead to high attrition rate and migration to the competitive platforms.

We have been able to achieve great results in our research. With our research results, it becomes convenient for Google to detect spam apps.

As mentioned above, there are 2.8 million apps on the app-store. Out of these 180K are categorized as entertainment apps. Based on the findings of the research paper we referred to, we know that around 3% of the total apps were spam which totals to approximately 80K spam apps. Finally, based on the association rules result we found 35K spam apps in the entertainment category. Hence, now if Google wants to detect the spam apps, they would not have to parse through the entire app-store but using our results, there are high chances of the app in Entertainment category being a spam.

Data Description

Data source: The data has been taken from an Individual research paper. The link where the data was present is: http://www.privmetrics.org/publications/www15/

Metadata:

Data Fields	Description	Data type
APPID	App identifier of the app. This is the unique identifier of the app to uniquely identify it in Google Play Store.	factor
APPNAME	Name of the app.	factor
CATEGORY	Category out of 30 Google Play Store Categories (In March 2014 Google increased the number of app categories. This data was collected before that).	factor
PRICE	Price of the app in USD.	numeric
CONTENT_RATING	Rating of the content hosted by the app.	factor
DOWNLOADS	Number of downloads of the apps.	factor
DOWNLOAD_MIN	Minimum number of downloads	factor
DOWNLOAD_MAX	Maximum number of downloads	factor
SIZE	Size of the app binary.	factor
CURRENT_VERSION	Version number given by the app developer.	factor
MIN_REQUIRES_ANDROID	Minimum Android version required to run the app	factor
MIN_REQUIRED_ANDROID_FIRST	Minimum Android version required to run the app	factor
UPDATED	Last updated date of the app.	factor
TOTAL_REVIEWS	Total number of users reviewed the app.	integer
AVERAGE_RATING	Average star rating of the app.	numeric
5_RATING	Number of users reviewed the app with 5 stars.	integer
4_RATING	Number of users reviewed the app with 4 stars.	integer
3_RATING	Number of users reviewed the app with 3 stars.	integer
2_RATING	Number of users reviewed the app with 2 stars.	integer
1_RATING	Number of users reviewed the app with 1 stars.	integer
DEVELOPER_SITE	URL of the developer's	factor
DEVELOPER_CONTACT	SHA1 hash of developer's email address.	factor
DEVELOPER_NAME	SHA1 hash of developer's name.	factor

spam	There are 551 apps labelled as spam. The rest of the apps are other apps by the developers of the labelled apps.	integer

The screenshot of the dataset is attached in the (Appendix A.a).

Why is this data of interest?

With technological advancement in mobile computing, mobile apps have become an integral part of this environment. With millions of floating apps and thousands of apps being released daily, spam apps have become a major threat to data privacy as well as to the user experience. It is equally painful for the businesses and users to deal with. During our process to find dataset for the project this dataset caught our attention. The researchers who collected it, worked on predicting a spam app at the point of entry. As we realized that already thousands of spam apps floating around on the android platform, we indigenously created our problem statement to predict spam apps already present on the platform. This dataset with substantial details about app performance on the store became our clear favorite. It had customer ratings, downloads, price, app category etc., which we felt had meaningful relevance to predict an app to be spam, and henceforth we decided to explore this dataset to reach our objective mentioned before.

Research Question

We categorized our business problem into parts which will be the questions, Google would want to answer when they want to remove the spam apps. The first part would be the amount of monetary loss which Google would have to let go by removing the spam apps. The second part, which we identified as our business problem is the credibility loss which Google faces when they feature a spam app. App users are very sensitive and one bad experience leads to un-installing an app. Under such a scenario the credibility loss outweighs the monetary loss.

As Google doesn't check the application against its explicit spam app policy during the time app is introduced in the market. It considers the app only after the customer ratings. This action is reactive in nature and introduces a considerable time lag between app submission and detection of spam behavior. Our project is doing the same thing but helping in finding the spam apps quicker. We are considering user ratings along with other factors which are useful in predicting the spam app. To predict whether the mobile application introduced in the market is a spam app or not. Considering the category and content of the application and the ratings it has got from the customers we are classifying the application as spam or not. Whether detecting the spam app quicker in time will help Google in providing better user experience and increase the credibility of application available in Google store. The revenue generated with spam apps can reduce and the revenue from paid non-spam apps will increase due to better customer satisfaction and trust in the Play store.

Methodology

Data Cleaning

Before we started with deploying any of the data modeling techniques, we cleaned the dataset by removing any occurrence of 'space' and special characters. Since the downloads were in range, we created two new columns named "DOWNLOADS_MIN" and "DOWNLOAD_MAX". Then after initial exploratory analysis we found that there were no spam apps above DOWNLOAD_MIN than 5 million, hence we removed all the rows above that mark. (Appendix B.a)

Also, we cleaned the MIN_REQUIRED_ANDROID by checking the app version names and replacing them with the number release for some of the fields and then took the first value in the MIN_REQUIRED_ANDROID field to create a new field called MIN_REUIRED_ANDROID_FIRST.

Exploratory Analysis

We started off with exploratory analysis using Tableau to find out the impact of variables on predictor column (spam) and on each other. (Appendix B.b, B.c).

Sampling Data

As our data set was unbalanced in the sense that the ratio of spam apps to the non-spam apps was very low. To train our models better, we deployed under-sampling and over-sampling of the data set. By this we could achieve more realistic accuracy and sensitivity which will be discussed later in the paper.

Data Models

To achieve the result that of finding the best spam app, we deployed both supervised and unsupervised learning. We started with supervised learning and implemented the logistic regression model, Ensemble methods of Bagging, Boosting and Random Forest, K-nearest neighbors and Classification trees.

i. Logistic Model: This is one of the simplest models for data mining and after plotting a heat map (Appendix B.d). Logistic regression predicts the probability of occurrence of an event by fitting data to a logit function.
ii. Classification Trees: We implemented both the normal and pruned tree to classify the spam apps.
iii. Ensemble Methods: They combine the results from multiple models with the goal of improving prediction accuracy. We deployed Bagging, which we found helped in improving the stability and accuracy of the models by giving us more accurate results. By applying Random Forest algorithm, we could correct the results of the decision trees which generally overfit on the training data set. Boosting helped us to reduce bias or variance.
iv. K-nearest neighbors: is one of the simplest supervised learning algorithms where prediction is made based out of approximated local computation which was very necessary in our case as it gave another dimension to solve our problem.

v. Association Rules: Amongst the unsupervised learning, this data mining technique could predict the best result which aligned to exploratory analysis results.

Results and Findings

As our model selection should be able to most precisely classify the spam apps which is done by measuring accuracy (Total number of apps correctly classified/ Total number of apps) and Sensitivity (total number of apps classified as spam/ (sum of apps correctly classified as spam and sum of apps which were spam but not classified as spam).

Attached below is a table of all the models that we run using sampled, under-sampled and over-sampled data.

Model	Sampling Types	Accuracy	Sensitivity
logistic	under-sampled	74.68	62.82
	over-sampled	83.9	37.82
bagging	sampled	91.4	5.17
	under-sampled	64.26	73.71
	over-sampled	89.3	16.02
Random-Forest	sampled	92.4	0
	under-sampled	67.58	74.35
	over-sampled	83.46	39.74
K-nearest	sampled	90.83	6.74
	under-sampled	61.25	43.72
	over-sampled	87.47	19.63
Classification Tree	sampled	93.19	0
	under-sampled	61.78	73.07
	over-sampled	69.28	63.46

Please find attached the graphical representation of the accuracy and sensitivity of all the supervised learning performed. (Appendix C.a)

Additionally, we performed basic unsupervised learning by applying Association Rules to determine the quintessential factors of a spam app. We identified top 5 factors which may be similar in most of the spam apps. The results are shown below.

```
    lhs                                                    rhs      support    confidence lift
[1] {CATEGORY=Entertainment,MIN_REQ_ANDROID_FIRST=2}    => {spam=1} 0.01138743 0.1847134  2.561180
[2] {CATEGORY=Entertainment}                            => {spam=1} 0.01295812 0.1672297  2.318757
[3] {CATEGORY=Personalisation,CONTENT_RATING=LowMaturity} => {spam=1} 0.01007853 0.1392405  1.930667
[4] {CONTENT_RATING=LowMaturity,MIN_REQ_ANDROID_FIRST=2} => {spam=1} 0.02460733 0.1081081  1.498994
[5] {CONTENT_RATING=LowMaturity}                        => {spam=1} 0.02997382 0.1057248  1.465949
```

Conclusion

Though we achieved high accuracy in some models but they were not important to us as they were not predicting the spam apps. For our prediction to be useful to business we need to have a trade-off between accuracy and sensitivity. The sensitivity will give the measure of the spam apps predicted correctly which is crucial for the business. Considering this trade-off Random Forest with under samples data and Classification tree with under sampled data are the two important models for our project.

The Association rule will help in weeding out spam apps quickly, as they would focus the search on the group of apps with higher probability of being spam, and then supervised learning models can be implemented to find out the spam apps.

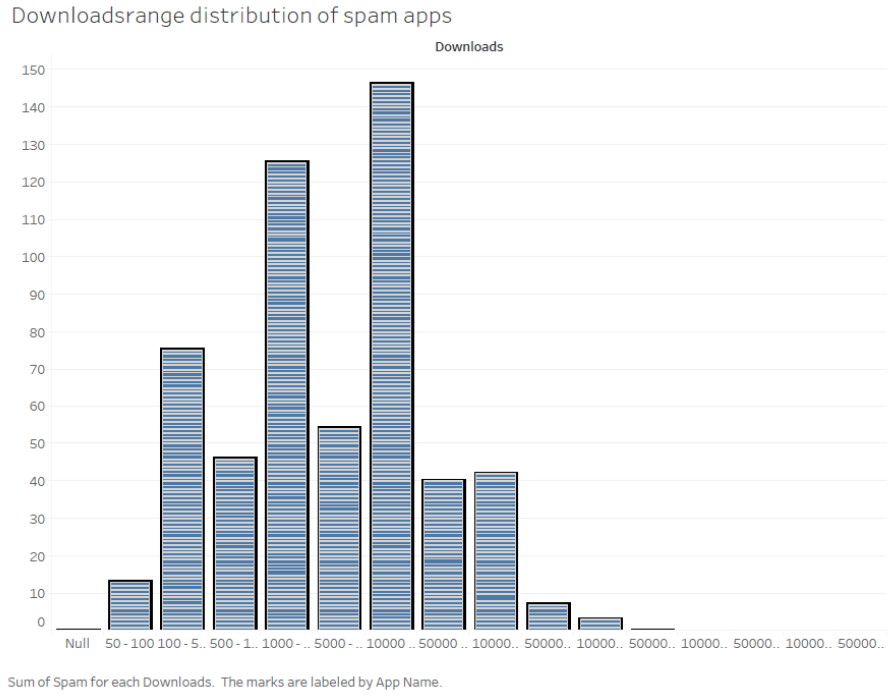
Appendix

A. Screenshot of the dataset

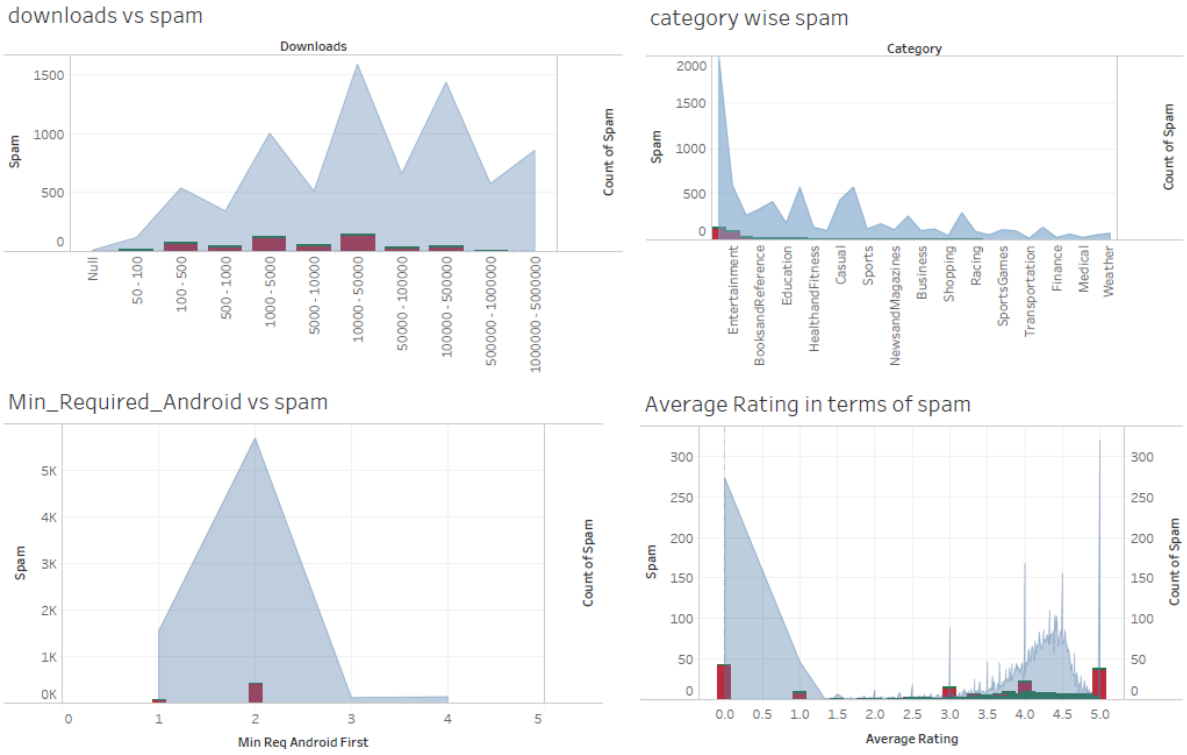
APP_ID	APP_NAM	CATEGOR	PRICE	CONTENT	DOWNLO	DOWNLO	DOWNLO	SIZE_MEG	CURRENT_	MIN_REQ	MIN_REQ	LASTUPDATED	TOTAL_RE	AVERAGE_	5RATING	4RATING	3RATING	2RATING	1RATING	D
FUNCOM.	Sexy Serie	Personalis	1.09	MediumMaturity	0	0	0.78	1	2.1	2	10/4/2012	0	0	0	0	0	0	0	h
FUNCOM.	Sexy Serie	Personalis	1.11	MediumMaturity	0	0	0.65	1	2.1	2	10/4/2012	0	0	0	0	0	0	0	h
com.cond	How To Lc	Healthanc	1.16	LowMaturity	0	0	3.4	1.4.6.56		2.2	2	4/11/2013	0	0	0	0	0	0	0	h
com.jb.go	GO SMS - I	Personalis	2.19	Everyone	0	0	4.8	1	2.2	2	27-11-2013	0	0	0	0	0	0	0	h
com.livew	Halloweer	Personalis	1.15	Everyone	0	0	3.4	1	2.2	2	10/10/2013	0	0	0	0	0	0	0	h
com.mobi	MSDict En	Booksand	1.99	Everyone	0	0	0.63	3.2.100		2	2	14-03-2012	0	0	0	0	0	0	0	h
dating.sec	Dating Se	Lifestyle	0.99	HighMaturity	0	0	5.4	1	2.2	2	23-01-2013	0	0	0	0	0	0	0	h
ecowork.p	pokCircle	Business	11.19	LowMaturity	0	0	1.2	2.00.000		2.2	2	11/12/2012	0	0	0	0	0	0	0	
air.Valent	Valentine	Entertainr	0	Everyone	50 - 100	50	100	0.14	1.0.0		2.2	2	5/2/2012	2	5	2	0	0	0	0	h
appinvent	Setting Gc	Booksand	0	Everyone	50 - 100	50	100	1.7	1	1.5	1	18-09-2013	0	0	0	0	0	0	0	h
au.com.de	Animal Ice	Personalis	0	LowMatur	50 - 100	50	100	2.6	1.3.0		2.1	2	13-04-2013	0	0	0	0	0	0	0	h
au.com.de	Big Planet	Personalis	0	LowMatur	50 - 100	50	100	2.3	1.3.0		2.1	2	13-04-2013	0	0	0	0	0	0	0	h
au.com.de	Galaxy No	Personalis	0	LowMatur	50 - 100	50	100	2.4	1.3.0		2.1	2	4/4/2013	0	0	0	0	0	0	0	h
au.com.de	View Live	Personalis	0	LowMatur	50 - 100	50	100	2.7	1.2.9		2.1	2	26-02-2013	0	0	0	0	0	0	0	h
au.com.de	Flow Live	Personalis	0	LowMatur	50 - 100	50	100	2.8	1.3.0		2.1	2	20-03-2013	0	0	0	0	0	0	0	h
au.com.de	Galaxy S4	Personalis	0	LowMatur	50 - 100	50	100	2.7	1.3.0		2.1	2	17-07-2013	0	0	0	0	0	0	0	h
au.com.de	IOS7 Gala	Personalis	0	LowMatur	50 - 100	50	100	2.3	1.3.0		2.2	2	22-07-2013	0	0	0	0	0	0	0	h
au.com.de	IOS7 Gala	Personalis	0	LowMatur	50 - 100	50	100	2.6	1.3.0		2.2	2	31-07-2013	0	0	0	0	0	0	0	h
au.com.de	Ocean De	Personalis	0	LowMatur	50 - 100	50	100	2.4	1.3.0		2.1	2	13-04-2013	1	1	0	0	0	0	1	h
au.com.de	Beach Gal	Personalis	0	LowMatur	50 - 100	50	100	2.4	1.2.9		2.1	2	27-02-2013	0	0	0	0	0	0	0	h
au.com.de	Red Cross	Personalis	0	LowMatur	50 - 100	50	100	2.3	1.2.9		2.1	2	26-02-2013	0	0	0	0	0	0	0	h
au.com.de	Teeth Live	Personalis	0	LowMatur	50 - 100	50	100	1.9	1.3.0		2.1	2	14-03-2013	0	0	0	0	0	0	0	h

B. Results of the exploratory analysis using tableau.

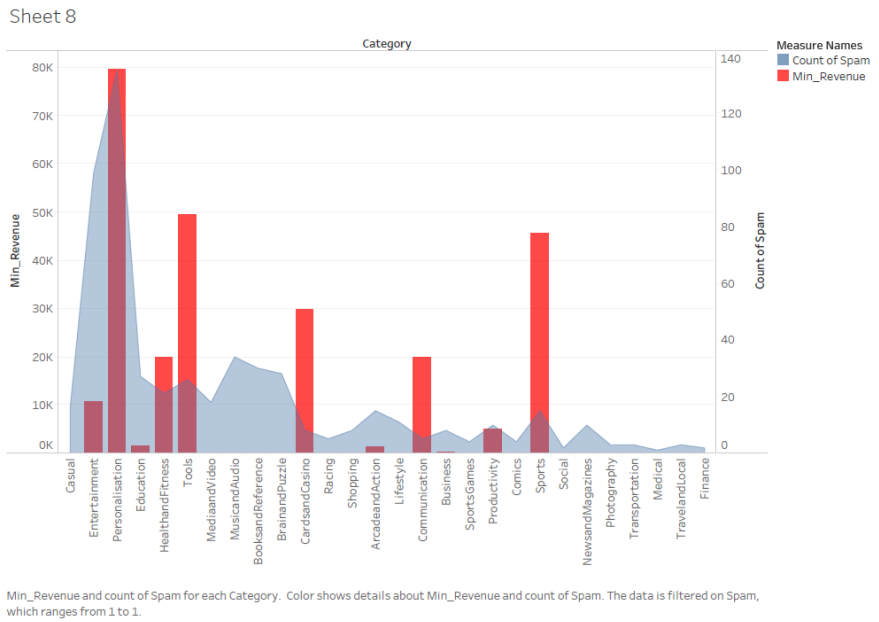
a. No spam apps above the range of 1 million



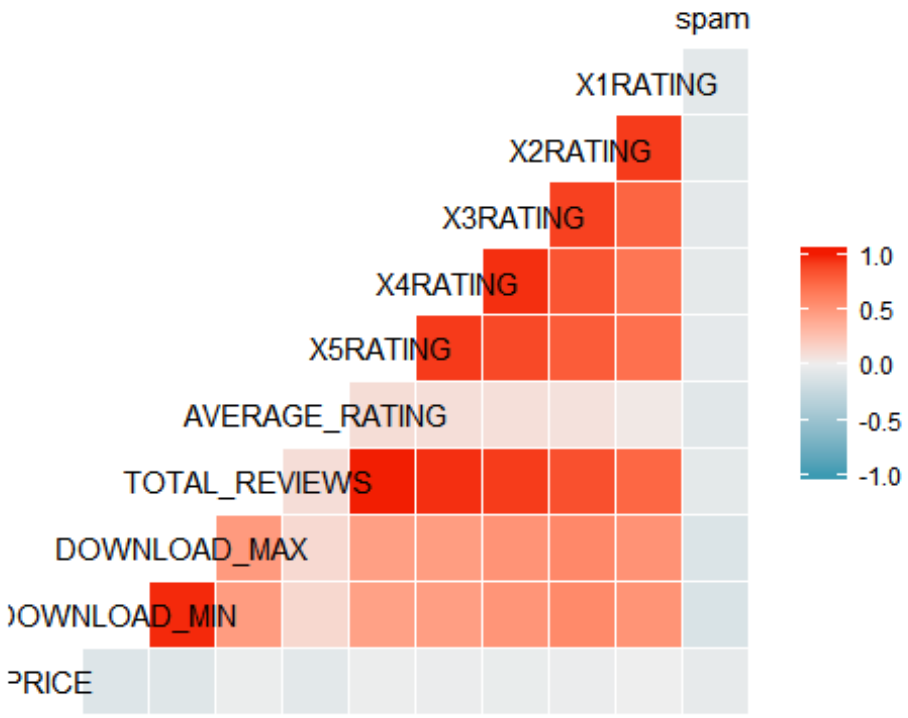
b. Number of spam based on major variables



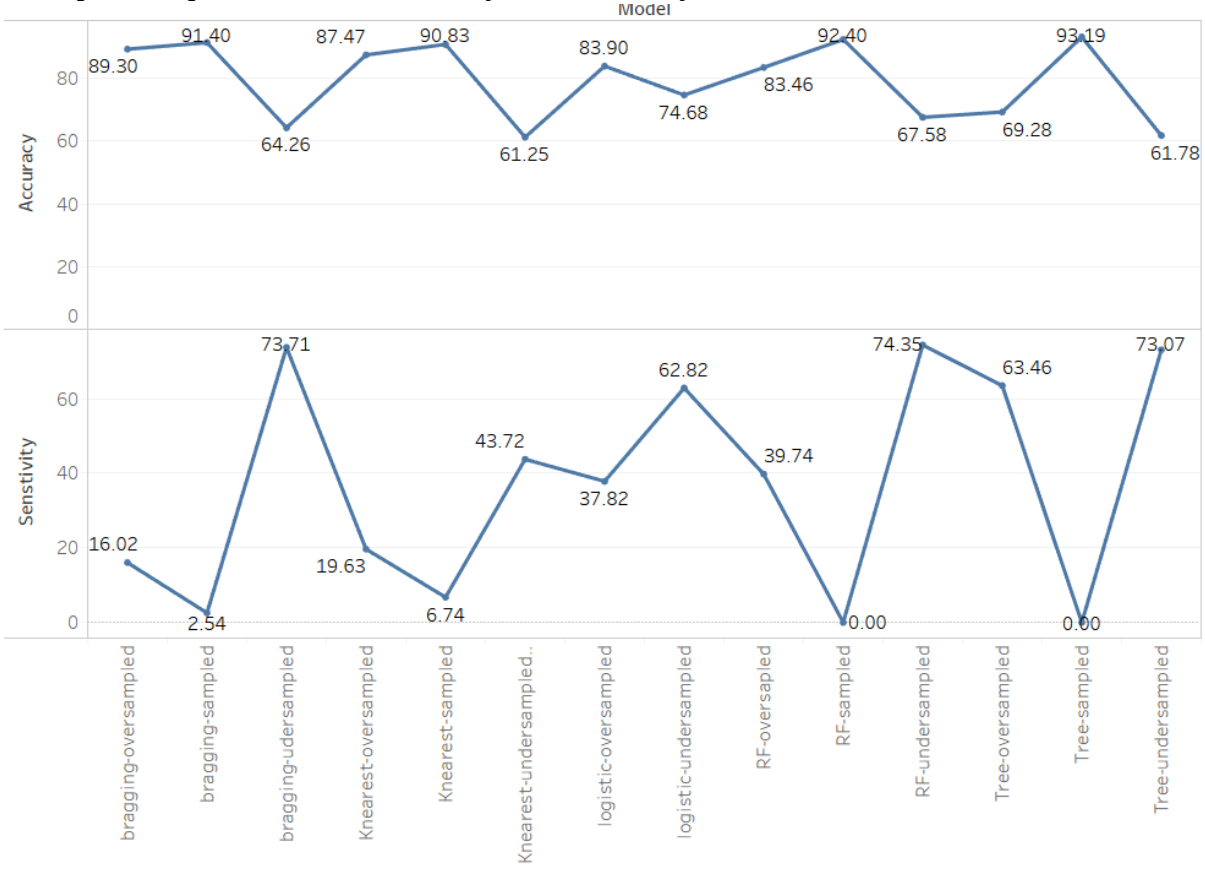
c. Profit earned by spam apps in terms of category



d. Heat map of all the variables to show the basic correlation amongst variables.



C. a. Graphical representation of accuracy and sensitivity.



References

https://en.wikipedia.org/wiki/Bootstrap_aggregating
https://en.wikipedia.org/wiki/Random_forest
https://en.wikipedia.org/wiki/Boosting_(machine_learning)
http://www.privmetrics.org/publications/www15/
https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
http://www.statsoft.com/Textbook/Association-Rules