Chat Application Success Prediction

Sunayna Ray[1], Dhanashri Patil[1]and Arnav Bhutani[1]

[1]VIT University, Vellore, Tamil Nadu, India

ABSTRACT
We find that any developer would like to predict the success rate of his application and abide by the rules that define the same. Thus we have collected the data from google playstore ourselves in an excel file. We have then made changes to the file to make it in arff format. Finally, have used WEKA tool for classification and association rule mining on an indigenously created dataset to get a decision tree for the aforementioned problem and to find the 10 best association rules on the topic. We are measuring the success of the app by the number of downloads on Google Play Store.
***Index Terms*—WEKA: Waikato Environment for Knowledge Analysis, Chat application, App Success, Predictor, Apriori, Decision tree, Arff, , Csv. , Dataset.**

I. INTRODUCTION

This is the era of internet and mobile phones. Chat Applications are a major part of this digital revolution. Gone are the days of typing elaborate emails (or even letters) for communicating with people far off. With this background, chat applications have proven to be of immense success.
Thus there is a large pool of customers for using such applications. Subsequently, there is a large pool of developers (and apps) too. Hence we find that any developer would like to predict the success rate of his application and abide by the rules that define the same. Here, we are measuring the success of the app by the number of downloads on Google Play Store.

PROBLEM STATEMENT

Due to the presence of numerous chat applications, the success of one's applications becomes critical. The demand is huge, but so is the supply. Any developer would wish to predict how successful his application will be and tailor its specifications accordingly. There are numerous apps on the Play Store, hence we have a high amount of data that can be mined to get the required knowledge.

II. DETAILED LITERARY SURVEY

Till date many research papers and projects have used Decision Trees as their method of classification. They can be used to predict GPA based on previous courses and to evaluate most important courses in their study plan [1]. They can also be used for Human Protein Function. Drug discoverers can easily use the model for predicting functions of proteins that are responsible for various diseases in human body [2].A credit card fraud detection problem for the resolution of reducing the bank's risk using decision tree algorithm has been proposed [3]. With the historical profile patterns, make use of credit card fraud detection models to equal the transaction information to predict the probability of being fraudulent for a new transaction. It offers a scientific basis for the authorization mechanisms.
To predict movie profitability a study using historical data on over 100 films produced in the United States (including their genre, opening month, duration, budget, etc. Decision trees are models commonly used in the field of artificial intelligence as decision support tools. The results show that the resulting model forecasts whether or not a movie will be profitable with an accuracy of over 70%, and this model can be used as a decision support tool for film producers [4]. Several efficiency recommendation system use decision trees. Decision tree classifiers like C4.5 and C5.0 algorithms have the merits of high accuracy, high classifying speed, strong learning ability and simple construction. In this paper, the decision-tree-based recommendation system framework is proposed. It uses efficient classification algorithm combined with collaborative recommendation approach for book recommendation. This hybrid book recommendation system combines advantages of both decision tree classifier and collaborative filtering. The results of C4.5 and C5.0 decision tree classifiers are compared and book recommendations are given to user by using efficient C5.0 decision tree classifier [5].

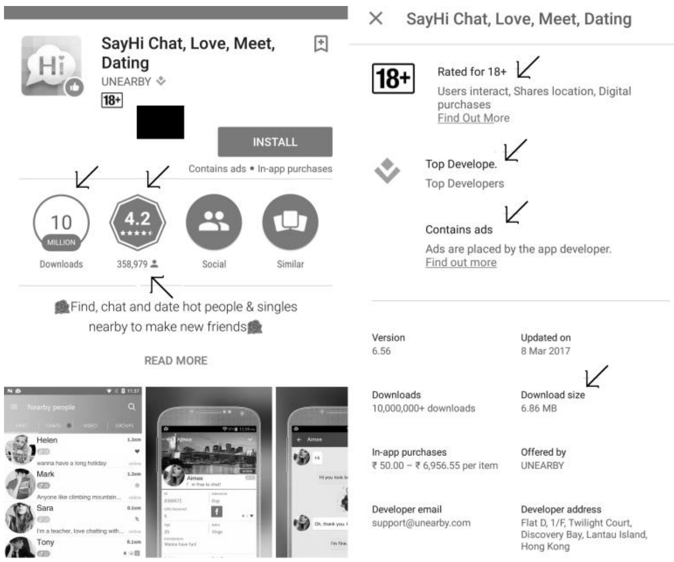
REFERENCED RESEARCH PAPERS:

1. Predicting Students Final GPA Using Decision Trees: A Case Study by Mashael A. Al-Barrak and Muna Al-Razgan
2. Human Protein Function Prediction using Decision Tree Induction By Manpreet Singh, Parminder Kaur Wadhwa and Parvinder Singh Sandhu
3. Credit Card Fraud Detection Using Decision Tree Induction Algorithm by Snehal Patil, Harshada Somavanshi, Jyoti Gaikwad, Amruta Deshmane, and RinkuBadgujar
4. Using Decision Trees to Characterize and Predict Movie Profitability on the US Market by María C. Burgos, María L. Campanario, Juan A. Lara, David Lizcano
5. Efficient Recommendation System Using Decision Tree Classifier and Collaborative Filtering by Sayali D. Jadhav1, H. P. Channe 2

III. DATA COLLECTION

The dataset has been collected from the Google Play Store. We have collected 9 attributes for 132chat apps. The attributes are:
Name
Minimum age
Top developer or not

Ads are allowed or not
Size
Rating
Number of 5 stars
Number of downloads
We have collected the above information from the following places (marked on pic):



ATTRIBUTES:

Name:
Name was collected to identify apps to avoid double mentioning.

Age:
Minimum age can affect the spectrum of audience to whom the app is available.

Top Developer:
If a person is a top developer then his apps are more likely to be shown above in the play store and google search results and an user using some other of his app may check the new one out too.

Advertisements:
Advertisements irritate the users and hence people are more likely to opt for apps without advertisements. On the other hand apps which provide the facility of showing advertisements may finance the developer so as to publicities the app itself.
Hence the affect of this attribute will be interesting to note.

Size:
The app size must affect its impact since users have memory constrain on their devices.

Rating of the app:
This shall given an idea to the new users about how the app is hence affect the success of the app.

No. of 5 stars:
This provides a proportion to find how many people actually rated the app.

IV. CONVERSION FROM DATA TO DATASET

Excel file:

	A	B	C	D	E	F	G	H	I
1	Sr.No	Name	Downloac	Age	Top Devel	Ads	Size (MB)	Rating	5 Stars
2	1	Woo	1	18+	No	Yes	21.54	4	
3	2	Glynk	0.1	12+	No	No	8.31	4.5	
4	3	Stranger c	0.1	18+	No	Yes	2.14	3.3	
5	4	Jaumo	10	18+	Yes	Yes	14.37	4.4	
6	5	Hi	10	18+	Yes	Yes	6.86	4.2	
7	6	WeChat	100	3+	Yes	No	34.09	4.2	
8	7	Nearby	1	12+	No	Yes	5.14	3.9	
9	8	InstaMess	10	12+	No	Yes	16.88	4.4	
10	9	Moco	10	18+	Yes	Yes	3.69	4.2	
11		Whatsapp	1000+		Yes	Yes	57	4.4	370+
12		Facebook	1000+		Yes	no	61.62	3.9	23+
13		Viber	500+		Yes	Yes	25.08	4.3	6.6+
14		IMO	100+		Yes	Yes	4.83	4.3	2+
15		Skype	500+		Yes	yes	20	4.1	5+
16		Telegram	100+		Yes	No	10.74	4.3	1+

From the details collected in Excel format the following changes were made:

- All the commas were removed.
- + signs were removed from all downloads values and everything was brought in terms of millions.
- + signs were removed from all 5 Stars values and everything was brought in terms of millions.
- 'k' for kb was removed from all Size values and everything was brought in terms of MBs.

Finally file was converted to CSV (Comma separated values) format:

Appcleaned.csv

```
Downloads (M),Age,Top Developer,Ads,Size (MB),Rati
1,18+,No,Yes,21.54,4
0.1,12+,No,No,8.31,4.5
0.1,18+,No,Yes,2.14,3.3
10,18+,Yes,Yes,14.37,4.4
10,18+,Yes,Yes,6.86,4.2
100,3+,Yes,No,34.09,4.2
1,12+,No,Yes,5.14,3.9
10,12+,No,Yes,16.88,4.4
```

Now to convert the CSV file to ARFF format (Attribute-Relation File Format) the syntax changes had to be added:

```
@relation Databasic

@attribute Sr.No numeric
@attribute Name {Woo,Glynk,'Stranger chat',Jaur
@attribute 'Downloads (M)' string
@attribute Age {18+,12+,3+,12}
@attribute 'Top Developer' {No,Yes,yes,no}
@attribute Ads {Yes,No,no,yes}
@attribute 'Size (MB)' string
@attribute Rating numeric
@attribute '5 Stars' numeric

@data
1,Woo,1,18+,No,Yes,21.54,4,?
2,Glynk,0.1,12+,No,No,8.31,4.5,?
3,'Stranger chat',0.1,18+,No,Yes,2.14,3.3,?
4,Jaumo,10,18+,Yes,Yes,14.37,4.4,?
5,Hi,10,18+,Yes,Yes,6.86,4.2,?
6,WeChat,100,3+,Yes,No,34.09,4.2,?
7,Nearby,1,12+,No,Yes,5.14,3.9,?
8,InstaMessage,10,12+,No,Yes,16.88,4.4,?
```

@relation Databasic

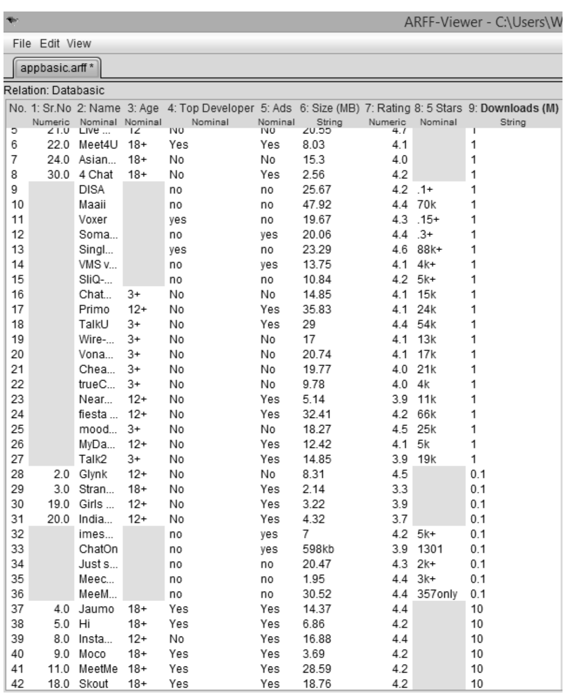
@attribute Sr.No numeric
@attribute Name {Woo,Glynk,'Stranger chat',Jaumo,Hi,WeChat,Nearby,InstaMessage,Moco ......' freePP,faceTalk,Wispi,Vimo,Tinder,Nearby-chat,Zalo,chatous,'fiesta by tango','Bee talk','ChaCha-video chat','text me',camfrog,tumblr,paltalk,'chat rooms','girls chat','text plus','message peeping Tom',W-match,SayHi,'waplog chat',badoo-meet,'mood message','verizon message',BetweenUs,MyDarling,ViMo,Talk2,'GMT talk','TU talk',Lycachat,TelTel,Voys,Vazii,menetalk}
@attribute 'Downloads (M)' string
@attribute Age {18+,12+,3+,12}
@attribute 'Top Developer' {No,Yes,yes,no}
@attribute Ads {Yes,No,no,yes}
@attribute 'Size (MB)' string
@attribute Rating numeric
@attribute '5 Stars' numeric

@data

It can be observed from the figure what input type has been attached to each attribute:

Name	Nominal
Minimum age	Nominal
Top developer or not	Nominal
Ads are allowed or not	Nominal
Rating	Numeric
Number of 5 stars	Numeric
Size	String
Number of downloads	String

Finally the ARFF file of the **dataset** was created:


ARFF-Viewer - C:\Users\W
File Edit View
appbasic.arff *
Relation: Databasic

No.	1: Sr.No Numeric	2: Name Nominal	3: Age Nominal	4: Top Developer Nominal	5: Ads Nominal	6: Size (MB) String	7: Rating Numeric	8: 5 Stars Nominal	9: Downloads (M) String
5	21.0	Live ...	12	No	No	20.55	4.7		1
6	22.0	Meet4U	18+	Yes	Yes	8.03	4.1		1
7	24.0	Asian...	18+	No	No	15.3	4.0		1
8	30.0	4 Chat	18+	No	Yes	2.56	4.2		1
9		DISA		no	no	25.67	4.2	.1+	1
10		Maaii		no	no	47.92	4.4	70k	1
11		Voxer		yes	no	19.67	4.3	.15+	1
12		Soma...		no	yes	20.06	4.4	.3+	1
13		Singl...		yes	no	23.29	4.6	88k+	1
14		VMS v...		no	yes	13.75	4.1	4k+	1
15		SliQ-...		no	no	10.84	4.2	5k+	1
16		Chat...	3+	No	No	14.85	4.1	15k	1
17		Primo	12+	No	Yes	35.83	4.1	24k	1
18		TalkU	3+	No	Yes	29	4.4	54k	1
19		Wire-...	3+	No	No	17	4.1	13k	1
20		Vona...	3+	No	No	20.74	4.1	17k	1
21		Chea...	3+	No	No	19.77	4.0	21k	1
22		trueC...	3+	No	No	9.78	4.0	4k	1
23		Near...	12+	No	Yes	5.14	3.9	11k	1
24		fiesta ...	12+	No	Yes	32.41	4.2	66k	1
25		mood...	3+	No	No	18.27	4.5	25k	1
26		MyDa...	12+	No	Yes	12.42	4.1	5k	1
27		Talk2	3+	No	Yes	14.85	3.9	19k	1
28	2.0	Glynk	12+	No	No	8.31	4.5		0.1
29	3.0	Stran...	18+	No	Yes	2.14	3.3		0.1
30	19.0	Girls ...	12+	No	Yes	3.22	3.9		0.1
31	20.0	India...	12+	No	Yes	4.32	3.7		0.1
32		imes...		no	yes	7	4.2	5k+	0.1
33		ChatOn		no	yes	598kb	3.9	1301	0.1
34		Just s...		no	no	20.47	4.3	2k+	0.1
35		Meec...		no	no	1.95	4.4	3k+	0.1
36		MeeM...		no	no	30.52	4.4	357only	0.1
37	4.0	Jaumo	18+	Yes	Yes	14.37	4.4		10
38	5.0	Hi	18+	Yes	Yes	6.86	4.2		10
39	8.0	Insta...	12+	No	Yes	16.88	4.4		10
40	9.0	Moco	18+	Yes	Yes	3.69	4.2		10
41	11.0	MeetMe	18+	Yes	Yes	28.59	4.2		10
42	18.0	Skout	18+	Yes	Yes	18.76	4.2		10

V. PRE-PROCESSING THE DATASET

This dataset has missing values since some of us maintained the columns of Sr. No and number of 5 star ratings and some of us didn't. Also the age value is available for few apps and unavailable for few.

Before:

Hence the following cleaning process was done:

Step 1. RemoveUseless:

Remove useless attribute: Name and serial number.
These have different value for each tuple.

Step2. Fill in missing data:

a. Few apps did not have age specification:
Default: 18+

b. Few of us noted down no. of 5 star ratings, few didn't: fill missing values with mean value.

Step3. Discretisation:

Convert Numeric value of Downloads attribute to nominal.

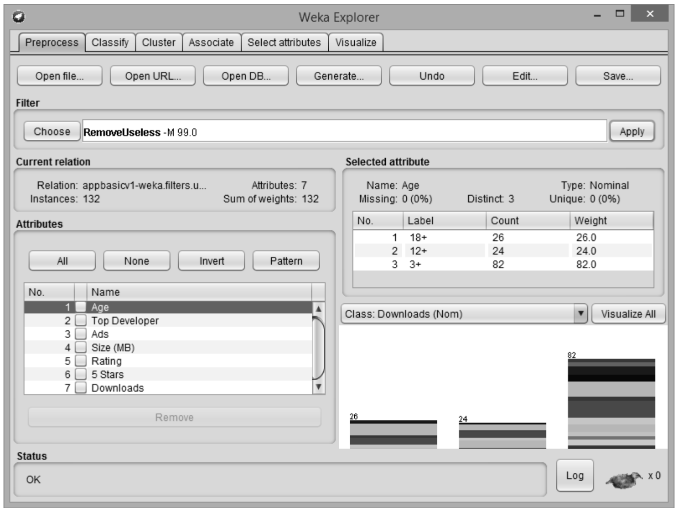
Step4. Normalise

Normalize all attributes except class attribute since that is discretised.
This is so that Apriori algorithm can be used.

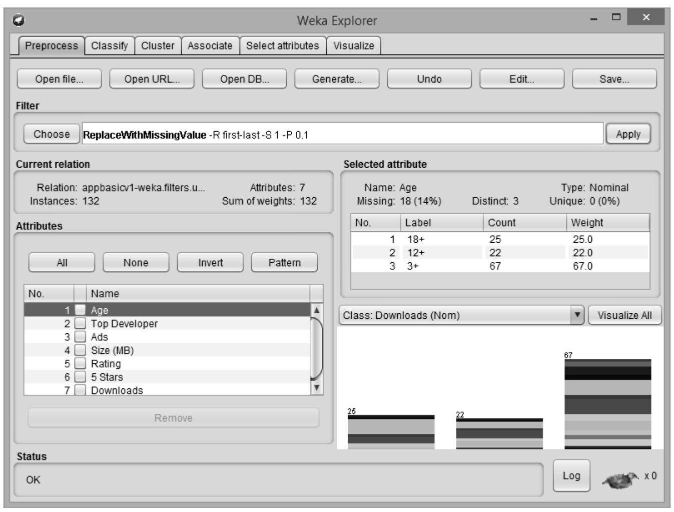
Step5. Merge infrequent nominal values in class attribute

FIGURES:

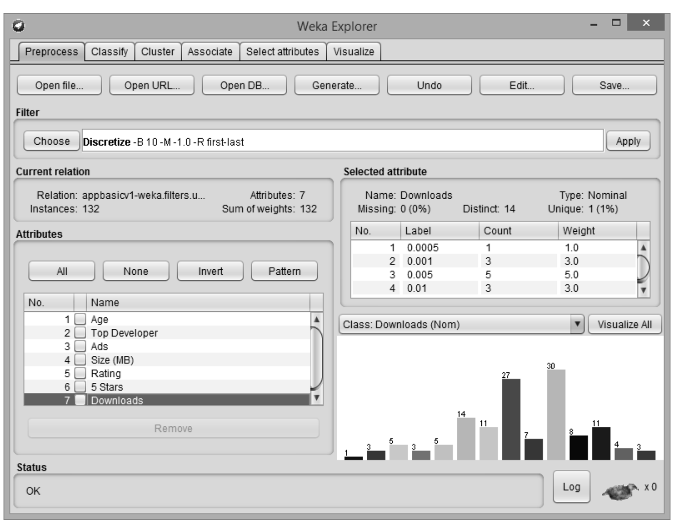
1) AfterStep 1:



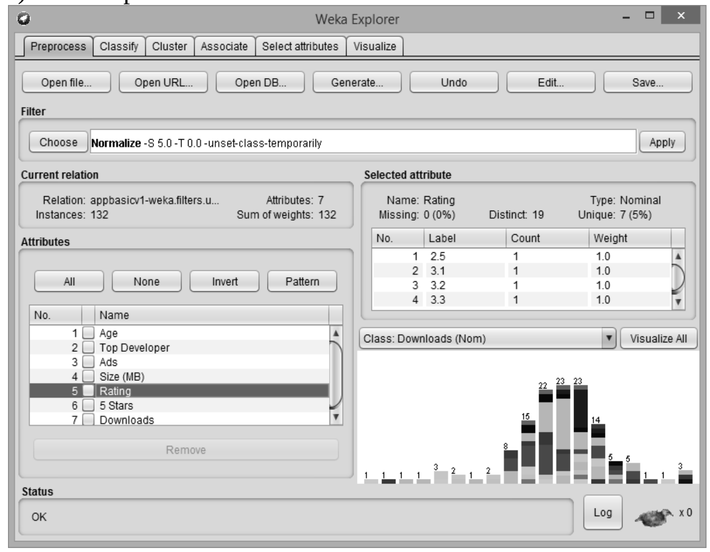
2) AfterStep 2:



3) After Step3:

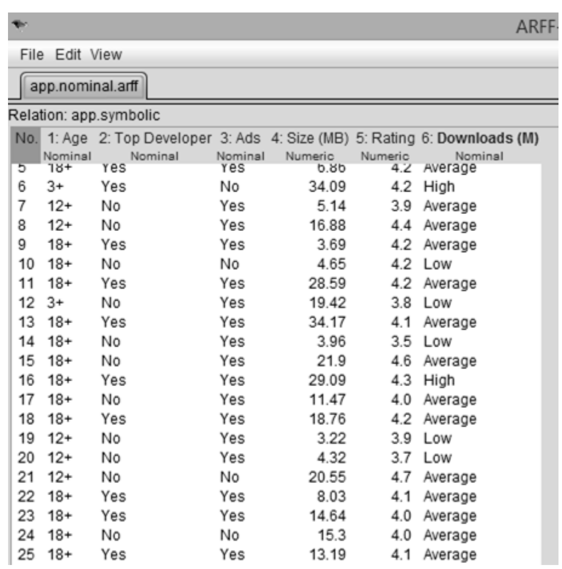


4) AfterStep 4:



5)AfterStep 5:

Thus the final dataset was:



ARFF
File Edit View
app.nominal.arff
Relation: app.symbolic

No.	1: Age (Nominal)	2: Top Developer (Nominal)	3: Ads (Nominal)	4: Size (MB) (Numeric)	5: Rating (Numeric)	6: Downloads (M) (Nominal)
5	18+	Yes	Yes	6.86	4.2	Average
6	3+	Yes	No	34.09	4.2	High
7	12+	No	Yes	5.14	3.9	Average
8	12+	No	Yes	16.88	4.4	Average
9	18+	Yes	Yes	3.69	4.2	Average
10	18+	No	No	4.65	4.2	Low
11	18+	Yes	Yes	28.59	4.2	Average
12	3+	No	Yes	19.42	3.8	Low
13	18+	Yes	Yes	34.17	4.1	Average
14	18+	No	Yes	3.96	3.5	Low
15	18+	No	Yes	21.9	4.6	Average
16	18+	Yes	Yes	29.09	4.3	High
17	18+	No	Yes	11.47	4.0	Average
18	18+	Yes	Yes	18.76	4.2	Average
19	12+	No	Yes	3.22	3.9	Low
20	12+	No	Yes	4.32	3.7	Low
21	12+	No	No	20.55	4.7	Average
22	18+	Yes	Yes	8.03	4.1	Average
23	18+	Yes	Yes	14.64	4.0	Average
24	18+	No	No	15.3	4.0	Average
25	18+	Yes	Yes	13.19	4.1	Average

VI. DATA MINING METHODOLOGY

TOOL USED

Weka:
Weka contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphicaluser interfaces for easy access to these functions.

Advantages:
- Free availability under the GNU General PublicLicense.
- Portability, since it is fully implemented in the Javaprogramming language and thus runs on almost any modern computing platform.
- A comprehensive collection of data preprocessing and modeling techniques.
- Ease of use due to its graphical user interfaces.

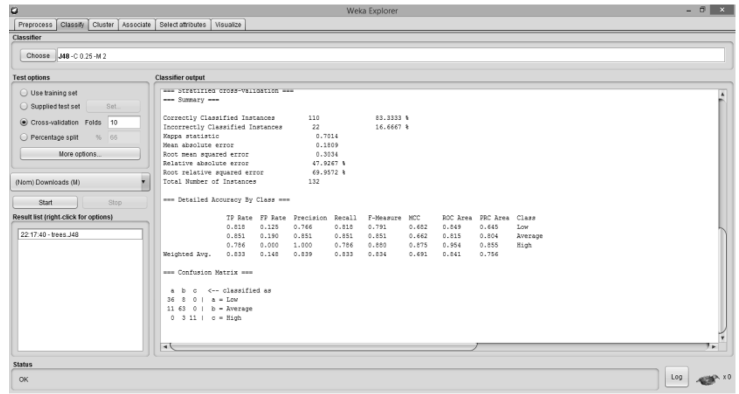
ALGORITHM USED

Decision Tree
A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm.A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules. They are simple to understand and interpret. People are able to understand decision tree models after a brief explanation. They have value even with little hard data. Important insights can be generated based on experts describing a situation (its alternatives, probabilities, and costs) and their preferences for outcomes. They allow the addition of new possible scenarios. Decision trees help determine worst, best and expected values for different scenarios.

Apriori Algorithm
Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.
Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as *candidate generation*), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k-1. Then it prunes the candidates which have an infrequent sub pattern

VII. RESULTS AND DISCUSSION

CLASSIFICATION
Decision Tree algorithm (J48) with 10 fold cross validation was used and the following was the result:



=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: app.symbolic-weka.filters.unsupervised.attribute.StringToNominal-R4-weka.filters.unsupervised.attribute.StringToNominal-R4-weka.filters.unsupervised.attribute.NumericToNominal-R4-weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-Rfirst-last

Instances: 132

Attributes: 6
Age
Top Developer
Ads
Size (MB)
Rating
Downloads (M)
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree

Top Developer = Yes
| Ads = Yes: Average (33.0/3.0)
| Ads = No: High (11.0)
Top Developer = No
| Ads = Yes: Low (47.0/11.0)
| Ads = No: Average (41.0/8.0)
Number of Leaves : 4
Size of the tree : 7
Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 110 83.3333 %
Incorrectly Classified Instances 22 16.6667 %
Kappa statistic 0.7014
Mean absolute error 0.1809
Root mean squared error 0.3034
Relative absolute error 47.9267 %
Root relative squared error 69.9572 %
Total Number of Instances 132

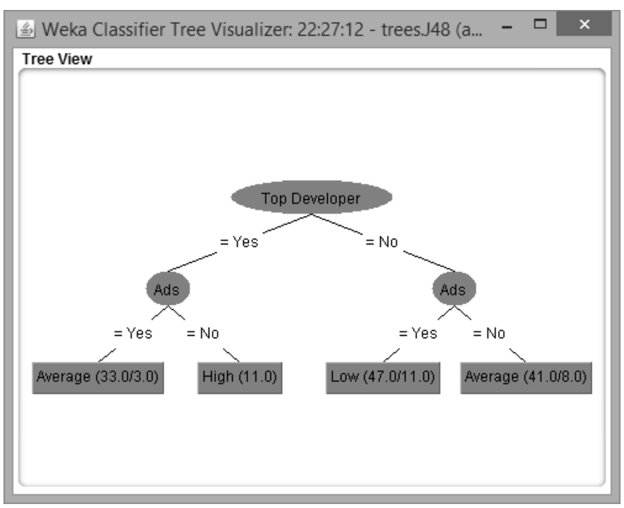
=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.818	0.125	0.766	0.818	0.791	0.682	0.849	0.645	Low
	0.851	0.190	0.851	0.851	0.851	0.662	0.815	0.804	Average
	0.786	0.000	1.000	0.786	0.880	0.875	0.954	0.855	High
Weighted Avg.	0.833	0.148	0.839	0.833	0.834	0.691	0.841	0.756	

=== Confusion Matrix ===

a	b	c	<-- classified as
36	8	0	| a = Low
11	63	0	| b = Average
0	3	11	| c = High

Tree:



What we found:

- 36, 63 and 11 Low, Average and High rated apps where classified correctly.
  8 Low rated apps where classified as Average.
  11 Average rated apps where classified as Low.
  3 High rated apps where classified as Average.
- We also find that that the main two defining attributes at higher level (since data set was compact – 132 tuples) are top developers and if advertisements are allowed or not.

ASSOCIATION RULE MINING

Apriori algorithm
This was to find useful association rules. The following was the result:

=== Run information ===

Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Relation: app.symbolic-

weka.filters.unsupervised.attribute.StringToNominal-R4-
weka.filters.unsupervised.attribute.StringToNominal-R4-
weka.filters.unsupervised.attribute.NumericToNominal-R4-
weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-Rfirst-last

Instances: 132

Attributes: 6
Age
Top Developer
Ads
Size (MB)
Rating
Downloads (M)

=== Associator model (full training set) ===

Apriori

=======

Minimum support: 0.1 (13 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 18
Generated sets of large itemsets:
Size of set of large itemsets L(1): 3
Size of set of large itemsets L(2): 40
Size of set of large itemsets L(3): 34

Size of set of large itemsets L(4): 9
Size of set of large itemsets L(5): 1

Best rules found:

1. Downloads (M)=Low 44 ==> Top Developer=No
44 <conf:(1)> lift:(1.5) lev:(0.11) [14] conv:(14.67)

2. Ads=Yes Downloads (M)=Low 36 ==> Top Developer=No
36 <conf:(1)> lift:(1.5) lev:(0.09) [12] conv:(12)

3. Ads=No Downloads (M)=Average 33 ==> Top Developer=No
33 <conf:(1)> lift:(1.5) lev:(0.08) [11] conv:(11)

4. Top Developer=Yes Downloads (M)=Average 30 ==> Ads=Yes
30 <conf:(1)> lift:(1.65) lev:(0.09) [11] conv:(11.82)

5. Age=18+ Top Developer=Yes Downloads (M)=Average 23 ==> Ads=Yes
23 <conf:(1)> lift:(1.65) lev:(0.07) [9] conv:(9.06)

6. Age=18+ Downloads (M)=Low 19 ==> Top Developer=No
19 <conf:(1)> lift:(1.5) lev:(0.05) [6] conv:(6.33)

7. Top Developer=Yes Rating='(4-4.25]' Downloads (M)=Average 17 ==> Ads=Yes
17 <conf:(1)> lift:(1.65) lev:(0.05) [6] conv:(6.7)

8. Top Developer=Yes Ads=Yes Rating='(4-4.25]' 17 ==> Downloads (M)=Average *17 <conf:(1)> lift:(1.78) lev:(0.06) [7] conv:(7.47)*

9. Age=18+ Ads=Yes Downloads (M)=Low 16 ==> Top Developer=No
16 <conf:(1)> lift:(1.5) lev:(0.04) [5] conv:(5.33)

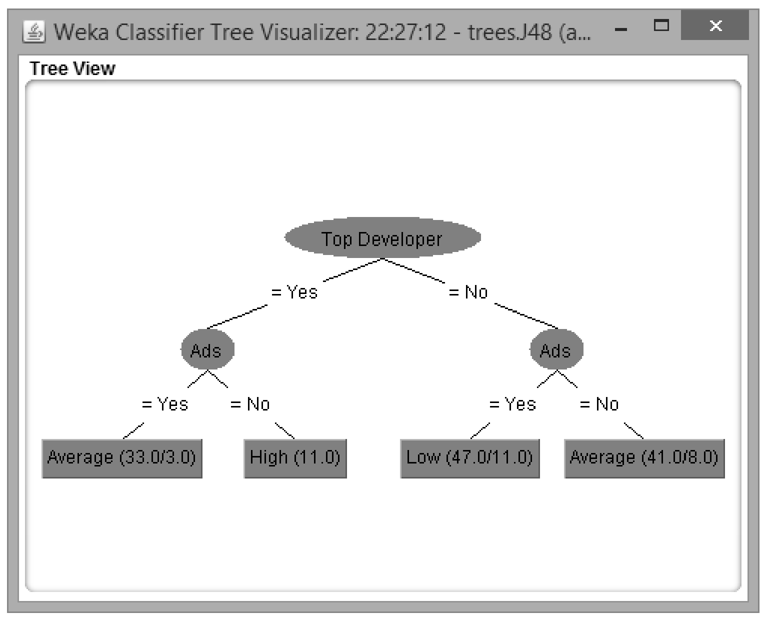
10. Downloads (M)=High 14 ==> Top Developer=Yes
14 <conf:(1)> lift:(3) lev:(0.07) [9] conv:(9.33)

What we found:

- If the downloads are low then most probably the developer is not a top developer.
- If the downloads are low and advertisements are allowed then most probably the developer is not a top developer.
- If the downloads are averageand advertisements are not allowed then most probably the developer is not a top developer.
- If the downloads are averageand developer is a top developerthen most probably advertisements are allowed.
- If the age limit is 18+, downloads are averageand developer is a top developerthen most probably advertisements are allowed.
- If the age limit is 18+, downloads are lowthen most probably developer is not a top developer.
- If the downloads are average, ratings are between 4 and 4.25and developer is a top developerthen most probably advertisements are allowed.
- If the advertisements are allowed, ratings are between 4 and 4.25and developer is a top developerthen most probably downloads are average.
- If the age limit is 18+, downloads are low and advertisements are allowed then most probably the developer is not a top developer.
- If the Downloads are high then most probably the developer is a top developer.

VIII. CONCLUSION

By mining the collected data successfully we can conclude upon the decision tree which can help determine the success of an app.



We also find an important association rule:
If the advertisements are allowed, ratings are between 4 and 4.25and developer is a top developerthen most probably downloads are average.

IX. FUTURE SCOPE

Application developers can in future use the concluded results to ensure that their developed application is successful. Furthermore, more relevant data can be collected on this topic and more accurate results can be found on the same.

X. REFERENCES

We have referred to the following materials and research papers:

1. Predicting Students Final GPA Using Decision Trees: A Case Study by Mashael A. Al-Barrak and Muna Al-Razgan

2. Human Protein Function Prediction using Decision Tree Induction By Manpreet Singh, Parminder Kaur Wadhwa and Parvinder Singh Sandhu

3. Credit Card Fraud Detection Using Decision Tree Induction Algorithm by Snehal Patil, Harshada Somavanshi, Jyoti Gaikwad, Amruta Deshmane, and RinkuBadgujar

4. Using Decision Trees to Characterize and Predict Movie Profitability on the US Market by María C. Burgos, María L. Campanario, Juan A. Lara, David Lizcano

5. Efficient Recommendation System Using Decision Tree Classifier and Collaborative Filtering by Sayali D. Jadhav1, H. P. Channe 2

6. J. Han and M. Kamber , Data Mining: Concepts and Techniques , Third Edition, Morgan Kaufman,2011.

7. I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Third Edition, Morgan Kaufmann, 2011.

8. G. K. Gupta, Introduction to Data Mining with Case Studies, Easter Economy Edition, Prentice Hall of India, 2014.

9. Alex Berson and Stephen J. Smith, Data Warehousing, Data Mining & OLAP, Tata McGraw Hill Edition, 2007.

10. Teli, S., & Kanikar, P. (2015). A survey on decision tree based approaches in data mining. International Journal of Advanced Research in Computer Science and Software Engineering, 5(4), 613-617.

11. Balasundaram, A., & Bhuvaneswari, P. T. V. (2013). Comparative study on decision tree based data mining algorithm to assess risk of epidemic.

12. Dai, Q. Y., Zhang, C. P., & Wu, H. (2016). Research of Decision Tree Classification Algorithm in Data Mining. International Journal of Database Theory and Application, 9(5), 1-8.

13. Al-Radaideh, Q. A., Al-Shawakfa, E. M., & Al-Najjar, M. I. (2006, December). Mining student data using decision trees. In International Arab Conference on Information Technology (ACIT'2006), Yarmouk University, Jordan.

14. Agarwal, S., Pandey, G. N., & Tiwari, M. D. (2012). Data mining in education: data classification and decision tree approach. International Journal of e-Education, e-Business, e-Management and e-Learning, 2(2), 140.

15. Venkatadri, M., & Reddy, L. C. (2010). A comparative study on decision tree classification algorithms in data mining. International journal of computer applications in engineering, technology and sciences (IJ-CA-ETS), 2(2), 24-29.

16.https://www.youtube.com/watch?v=LcHw2ph6bss&list=PLm4W7_iX_v4NqPUjceOGd-OKNVO4c_cPD