Appendices

A Exploratory Data Analysis

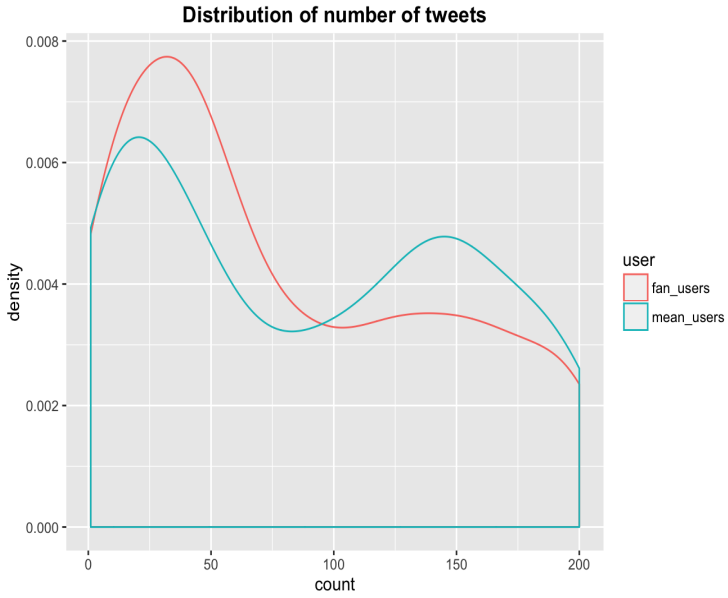

Figure 4: Distribution of number of tweets of each user

Sentimental analysis for each tweet Analyze sentiment of each tweet. Among all mean users' tweets, 22% are negative, and 67% positive. Among all fan users' tweets, 15% are negative and 76% are positive. Mean users overall tend to be more negative while less positive than fan users. The results are illustrated in figure 5.

Sentimental analysis for each person Analyze sentiment of each person. Integrate all tweets from each user as a single text and analyze the sentiment of this text. Among all mean users, 41% are negative and 18% are positive. While among fan users, 11% are negative and 47% are positive. The results are illustrated in figure 6. Again mean users overall tend to be more negative while less positive than fan users, and the difference is greater than tweet level sentimental analysis.

Sentimental percentage distribution for each person A user cannot be positive or negative all the time. To analyze how often a user being positive or negative, analyze sentiment of each tweets of that user, calculate how many percentage of positive and negative tweets among all tweets of that user. Plot the distribution of percentage positive and percentage negative in figure 7. Mean users overall tend to be more negative while less positive than fan users.

	words for anti-fans		words for fans
1	popcrave	1	happy
2	hillaryclinton	2	will
3	fuck	3	please
4	shit	4	always
5	musicnewsfacts	5	life
6	even	6	great
7	lol	7	hope
8	realdonaldtrump	8	back
9	never	9	best
10	potus	10	thank
11	better	11	taylornation13
12	fucking	12	beautiful
13	bitch	13	music
14	omg	14	never
15	life	15	follow
16	please	16	lol
17	god	17	miss
18	ill	18	song
19	man	19	ever
20	ever	20	birthday
21	stop	21	thanks
22	best	22	right
23	girl	23	night
24	happy	24	girl
25	hate	25	well
26	thank	26	even
27	yall	27	wait
28	ass	28	taylorswift
29	well	29	ive
30	video	30	video

Table 3: Top 30 frequent words used by anti-fans and fans

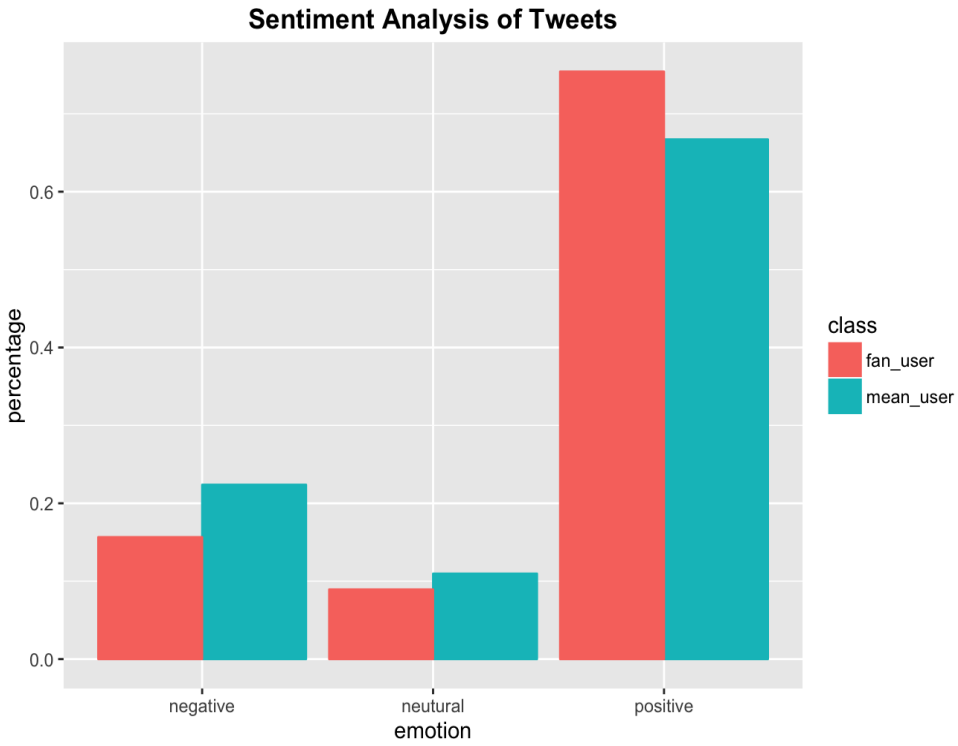

Figure 5: Sentimental analysis of tweet, with 25,915 tweets from 290 mean users, and 41,244 tweets from 510 fan users.

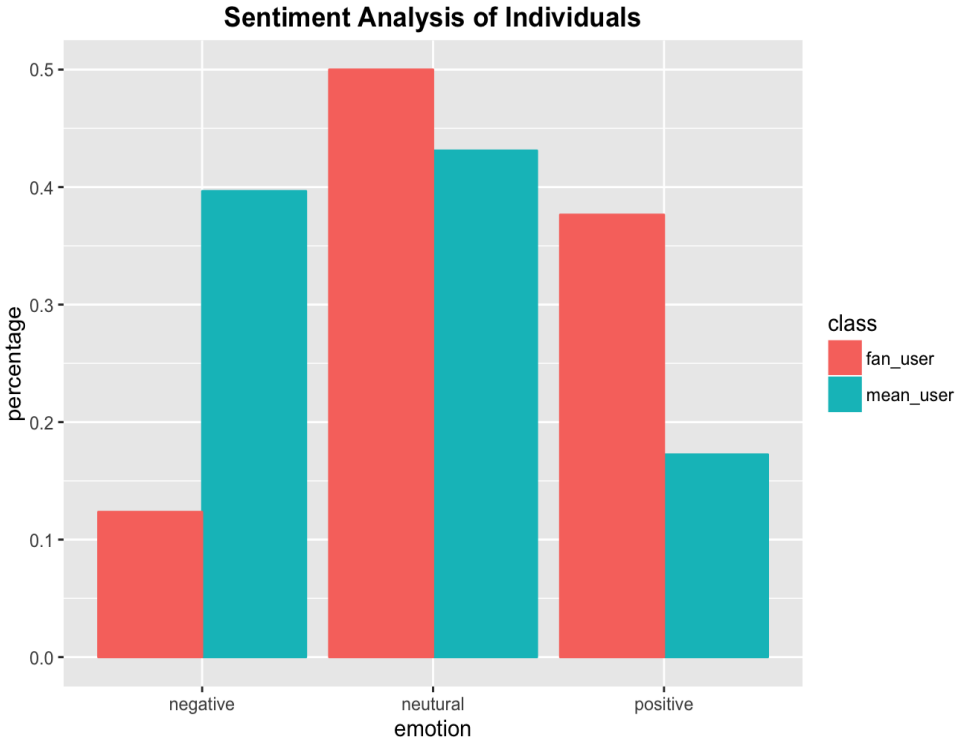

Figure 6: Sentimental analysis of Individuals, with 290 mean users and 510 fan users.

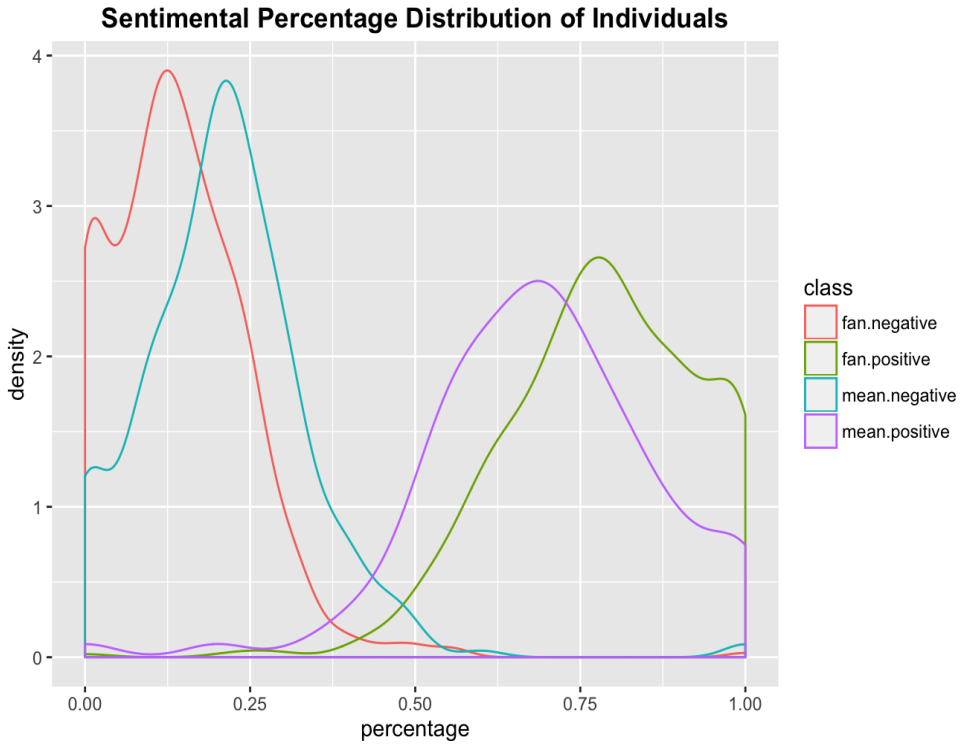

Figure 7: Sentimental percentage distribution of Individuals, with 290 mean users and 510 fan users.

B Statistical Implement and Results

B.1 Penalized Logistic Regression

Use the R package "glmnet" [11]. We set $\alpha = 1$ which is a L_1 penalty. To select best parameter λ, we use "cv.glmnet()" function to perform 10 fold cross validation on the training set. The optimal parameter is $\lambda = 0.02$. Words with non-zero coefficients are listed in Appendix table 4, rank by the absolute value of the coefficients in 4. The interpretation of the coefficient is that a unit change of the frequency of the corresponding word, the coefficient unit change of the logit probability of belonging to the mean user class. The positive coefficients indicate more frequently this word is used, more likely the user is an anti-fan. And the negative coefficients indicate more frequently this word is used, more likely the user is a fan. The larger the absolute value of the coefficient, the more influence of the corresponding word on the model.

B.2 Tree-based Methods

B.2.1 Classification Tree

Use R package "tree" [16]. We fit a classification tree with all 200 predictors on the training set. There are 25 variables used in building the tree with 28 terminal nodes. The misclassification rate for the training data is 0.09062.

The training error rate is very small, that is because classification tree tends to overfit the

(a) Words for mean users

	word	coef
1	black	147.188
2	free	143.895
3	sex	106.671
4	white	106.149
5	legend	105.455
6	shit	104.613
7	dick	101.984
8	ugly	98.706
9	bitch	90.163
10	jesus	87.066
11	snake	78.021
12	flop	72.958
13	beyonce	53.050
14	said	47.932
15	yall	35.644
16	know	24.973
17	ended	24.726
18	better	21.235
19	hillaryclinton	18.336
20	popcrave	8.551
21	lmao	8.293
22	fuck	5.541
23	pussy	4.701
24	take	4.260
25	bangyourankles	2.540
26	pay	1.341
27	fat	1.150
28	fun	0.961
29	like	0.760
30	queen	0.577
31	name	0.559
32	hot	0.471
33	hell	0.307
34	dont	0.208
35	one	0.059

(b) Words for fan users

	word	coef
1	proud	-107.394
2	friend	-54.694
3	ts6	-43.962
4	person	-42.679
5	love	-36.106
6	day	-32.580
7	smile	-32.048
8	lang	-31.866
9	haha	-24.621
10	awesome	-20.701
11	miss	-19.529
12	beautiful	-12.145
13	naman	-11.721
14	see	-9.968
15	tay	-9.962
16	amazing	-5.230
17	today	-3.773
18	yung	-3.257
19	live	-2.320
20	hope	-1.902

Table 4: Words selected by Penalized logistic Regression

data. We prune the tree by adding a penalty for the tree size. We use 10 fold cross validation to choose the best tree size. The best tree size is 14. The error rate on the training set is 0.1172.

The words used in construction the full and the pruned tree are listed in table 5. The full and the pruned tree are plotted in figure 8 and figure 9 with cutoffs of each node.

B.2.2 Bagging and Random Forest

Although we prune the tree to reduce over-fitting, the above tree based method still suffer from high variance, especially we have a larger number of predictors. So we further consider technics to reduce variance of our model. The most commonly used approach is bootstrap aggregation, or bagging. The idea is to take B bootstrap samples from the training set, fit a model on each bootstrap sample, then average over B samples to reduce variance. Denote the fitted model for the bth bootstrap sample as $\hat{f}^b(x)$, then the average bagging prediction is

$$\hat{f}(x) = \frac{1}{B}\sum_{i=1}^{B} \hat{f}^b(x)$$

In the classification case, we take the majority vote for the B predictions as our final prediction for the observation x. The tree constructed by each bootstrap sample is not pruned.

Random forest further improves the performance of bagging by decorrelating the trees. Instead of consider all possible splits for all predictors at each internal node, random forest randomly select m predictors to split on. Then taking the majority vote for B bootstrap trees gives the final prediction.

Use R package "randomForest" [15]. For bagging, we take 500 bootstrap samples and fit 500 trees, with number of variables tried at each split is m $=$ p $=$ 200. For random forest, we take 500 bootstrap samples and fit 500 trees, with number of variables tried at each split is $m = \sqrt{p} = 14$.

Averaging over multiple trees would be difficult to interpret the classification rules. Here we use varImpPlot() function to plot the importance predictors. The importance is evaluated essentially by the total amount that the misclassification rate or Gini index decreased due to splits over a given predictor, averaged over all B trees [13]. Variable importance plots are shown in figure 10 and figure 11.

(a) Words used in full classification tree

	Words
1	love
2	fuck
3	just
4	great
5	music
6	bitch
7	now
8	days
9	via
10	can
11	cute
12	take
13	que
14	follow
15	come
16	fan
17	taylorswift13
18	black
19	world
20	shawnmendes
21	name
22	snake
23	new
24	heart
25	like

(b) Words used in pruned classification tree

	words
1	love
2	fuck
3	bitch
4	take
5	que
6	follow
7	taylorswift13
8	black
9	world
10	days
11	shawnmendes
12	snake

Table 5: Words used in full and pruned trees

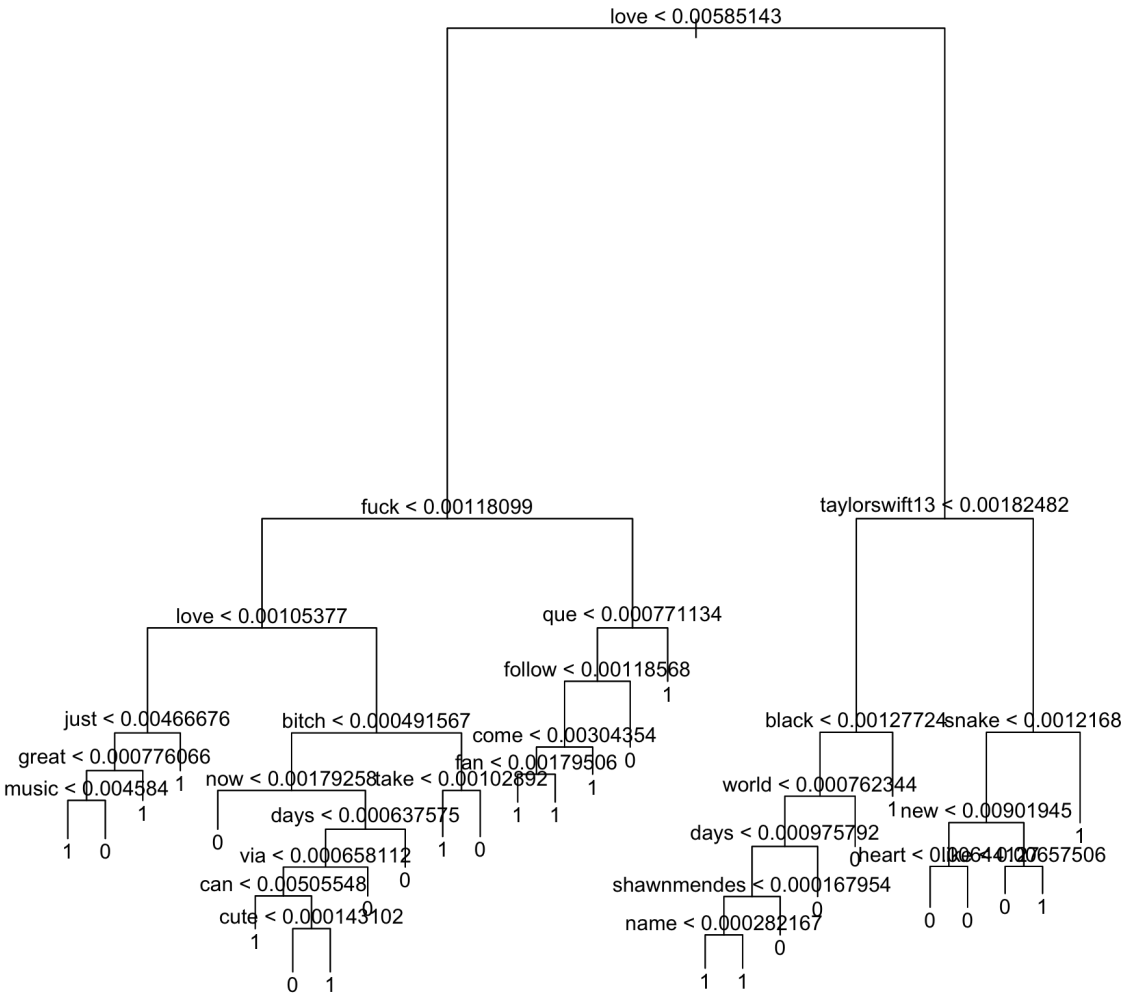

Figure 8: Full classification Tree

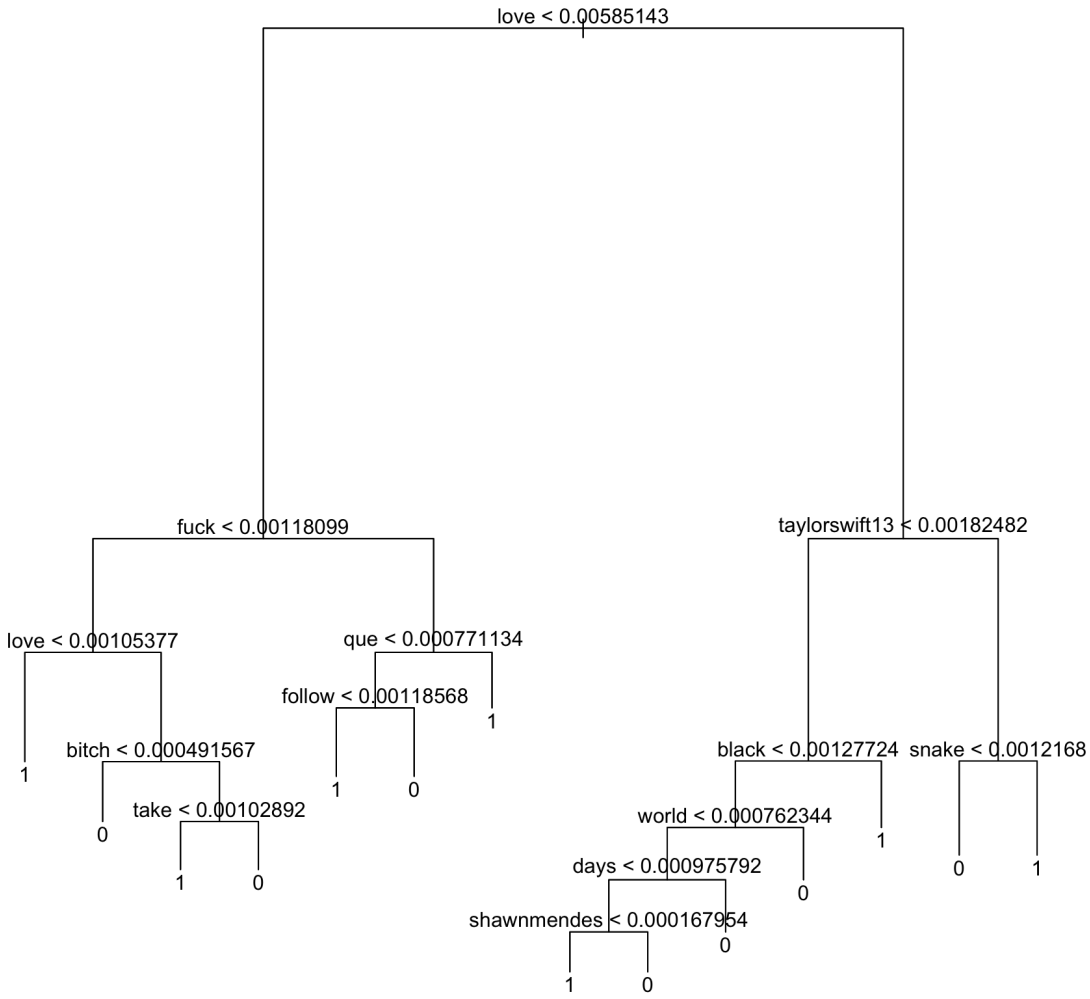

Figure 9: Pruned classification Tree

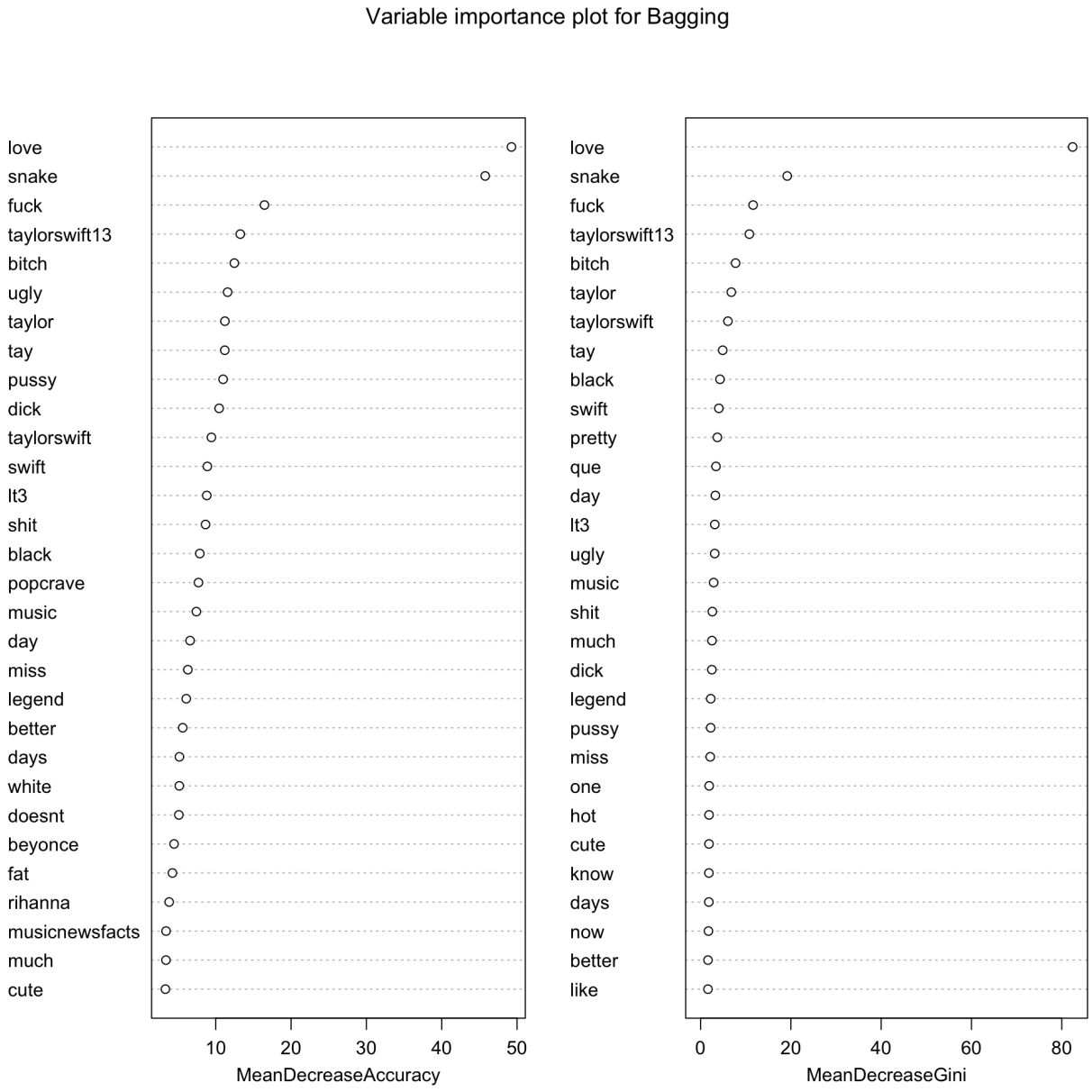

Figure 10: Variable Importance Plot for Bagging

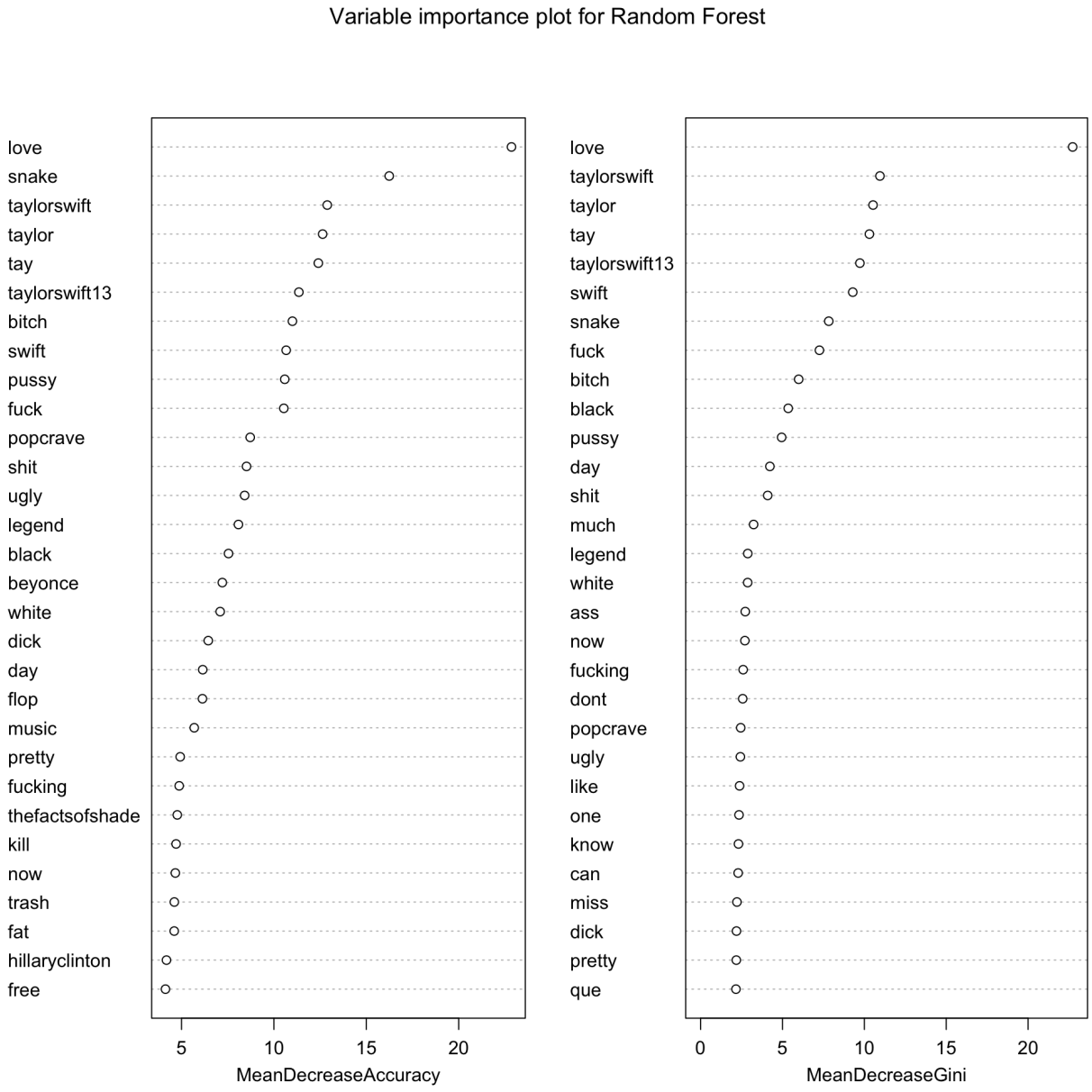

Figure 11: Variable Importance Plot for Random Forest

List of Figures

List of Tables