Gender Classification on Twitter dataset

Author: Nishant Salvi (nsalvi@umail.iu.edu)

Objectives and Motivation:

Objectives:
Through this project, I aim to accomplish and understand several objectives:

1. Highlight the importance of data preprocessing and feature engineering in algorithm performance.
2. Exploratory data analysis of users tweets, emotions, profile description and other features to highlight why gender classification through text is such a complex problem.
3. How well do words in tweets and profiles predict user gender?
4. What are the words that strongly predict male or female gender?
5. How well do stylistic factors (like link color and sidebar color) predict user gender?

Motivation:

- Gender classification based on text has always been an interesting problem and when you add even brands or franchises to that mixture the complexity of the problem escalates.
- I chose this project due to my sheer interest in Natural Language Processing, Machine Learning and Patterns in human behavior.
- Another aim was to highlight the importance of data preprocessing and feature engineering in algorithm performance.
- Also, I wanted to explore this interesting dataset to find patterns and correlations between different features.

The project can be divided into two parts respectively:

1 Exploratory data analysis
2 Training, testing and evaluation by different algorithms

Description of dataset:

The data at disposal looks as follows in raw format:
The dataset contains the following fields:

- **_unit_id**: a unique id for user
- **_golden**: whether the user was included in the gold standard for the model; TRUE or FALSE
- **_unit_state**: state of the observation; one of *finalized* (for contributor-judged) or *golden* (for gold standard observations)
- **_trusted_judgments**: number of trusted judgments (int); always 3 for non-golden, and what may be a unique id for gold standard observations
- **_last_judgment_at**: date and time of last contributor judgment; blank for gold standard observations
- **gender**: one of *male*, *female*, or *brand* (for non-human profiles)
- **gender:confidence**: a float representing confidence in the provided gender
- **profile_yn**: "no" here seems to mean that the profile was meant to be part of the dataset but was not available when contributors went to judge it
- **profile_yn:confidence**: confidence in the existence/non-existence of the profile
- **created**: date and time when the profile was created
- **description**: the user's profile description
- **fav_number**: number of tweets the user has favorited
- **gender_gold**: if the profile is golden, what is the gender?
- **link_color**: the link color on the profile, as a hex value
- **name**: the user's name
- **profile_yn_gold**: whether the profile y/n value is golden
- **profileimage**: a link to the profile image
- **retweet_count**: number of times the user has retweeted (or possibly, been retweeted)
- **sidebar_color**: color of the profile sidebar, as a hex value
- **text**: text of a random one of the user's tweets
- **tweet_coord**: if the user has location turned on, the coordinates as a string with the format "[*latitude*, *longitude*]"
- **tweet_count**: number of tweets that the user has posted
- **tweet_created**: when the random tweet (in the **text** column) was created
- **tweet_id**: the tweet id of the random tweet
- **tweet_location**: location of the tweet; seems to not be particularly normalized
- **user_timezone**: the timezone of the user

Exploratory data analysis (EDA):

Objective: In exploratory data analysis, I try to visualize the data using different charts, tables and formulae to get a fair amount of idea regarding the type of data I am dealing with. Also another motivation behind this is to get a fair idea of the dataset w.r.t. to the objectives.
Eg: If the dataset at disposal gives us features that have no distinct correlation with the class label (gender) then the performance of the algorithm is surely be unsatisfactory. EDA helps us in getting a beforehand idea of this phenomenon. One more thing I would like to add is, EDA is very useful for trying out your own hypothesis, so that you can tune your algorithm accordingly.

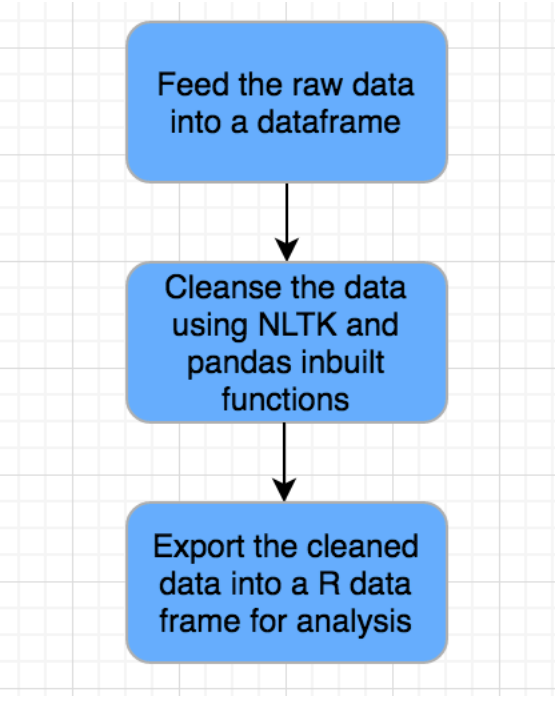
Technologies used:

1. R: Due to the beautiful inbuilt packages at disposal and its versatility
2. NLTK: For cleaning the dataset from stop words, non-English tweets, proper nouns.
3. Pandas framework: For its sheer flexibility and data manipulation and storing the data frame in pickle format which can be picked by a R script as well.
4. Several of the R libraries for visualization, data cleaning, vectorization, and data manipulation
5. Scikit learn: For TF-IDF and Count Vectorization, so that I get the most important words with respect to a particular tweet as well as the entire corpus

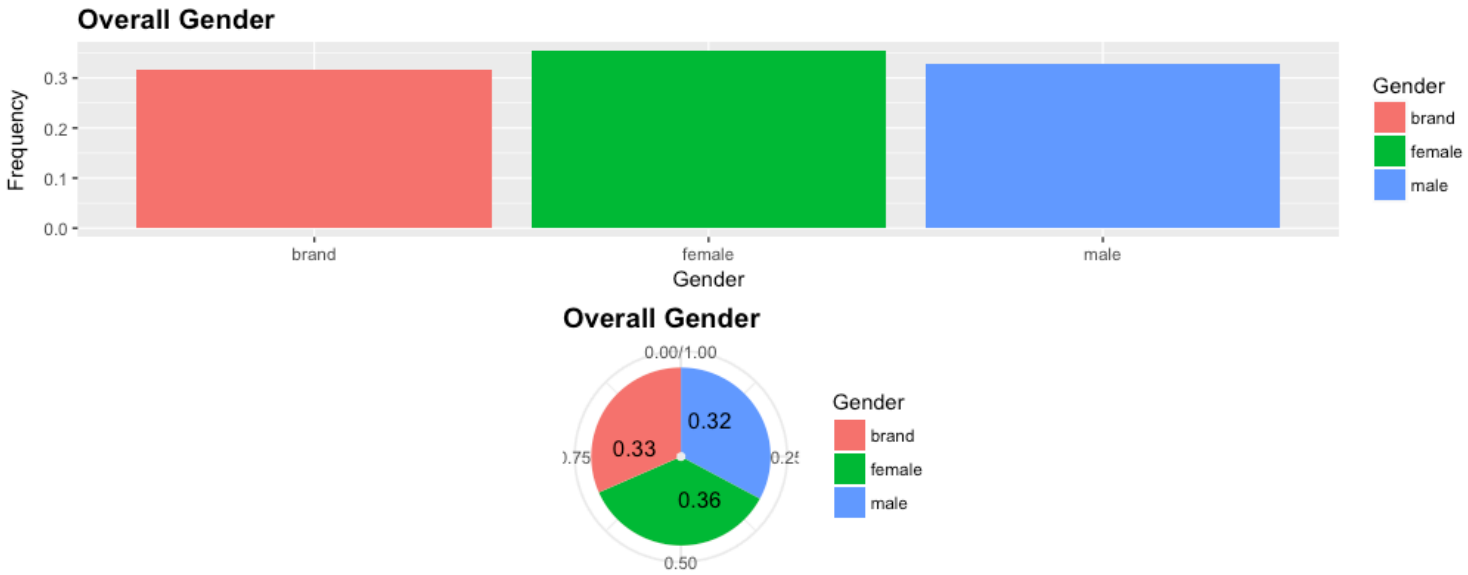
So basically, I have used Python for the data pre-processing part and R for further processing and visualization.

Pipeline:
Steps:

1. Picked the raw data into a Pandas dataframe. This dataframe is now a common intersection between EAD and algorithm application.
2. In this step, I removed rows which were blank, not in English and whose was either unknown or not specified. NLTK has built in library for checking the stopwords. These stopwords are matched with different languages and language ratios are calculated. If the max ratio corresponds to English, then only that particular row is considered.
3. I exported this semi-cleaned pandas dataframe and imported in my R script.



Note: All the below visualizations and tables have been generated through code which is attached separately and uploaded through Canvas.
Frequency chart: The goal of this visualization is to see whether the data is favoring a certain class label. Basically to get a visual glance at the dataset distribution w.r.t. to the different class labels:
Inference: The dataset is nicely balanced and does not lean to any particular gender.

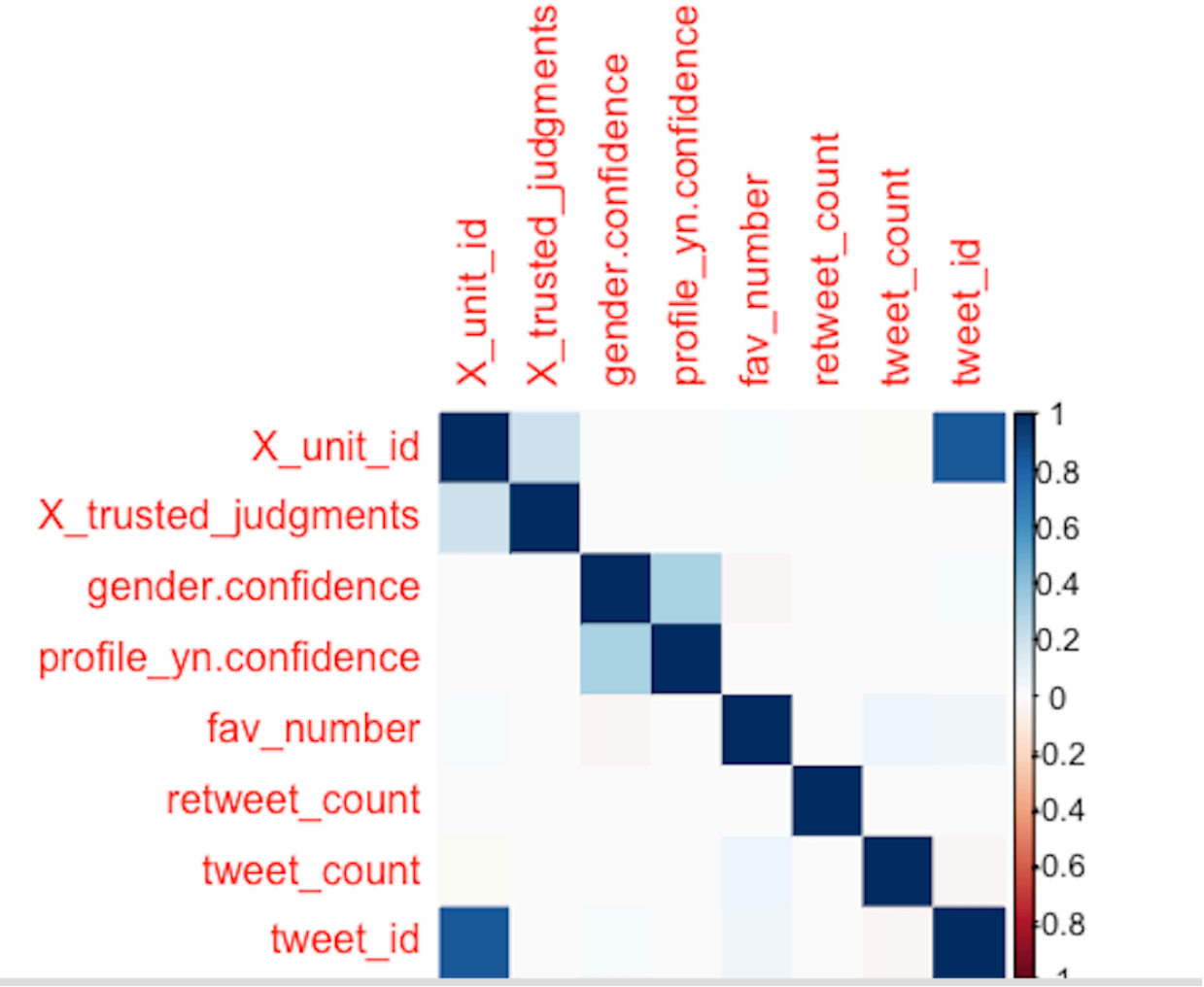


Feature Reduction and Combination:
Now that we know the dataset is balanced, it is necessary to find features that are really helpful. One reason for feature reduction is to take into account the quantity of distinct attributes the dataset is displaying for a given feature. For example:

	brand	female	male
FALSE	5928	6685	6173
TRUE	14	15	21

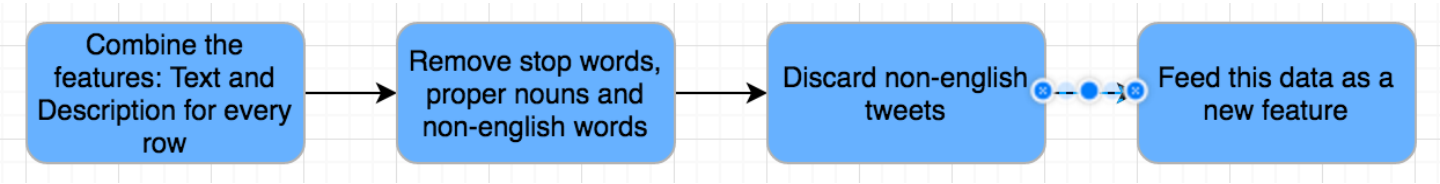
Above is a table generated in R corresponding to _golden profile feature. Very few users are displaying this feature and hence it has been omitted. Similarly, location and time zone analysis and its correlation with gender cannot be the basis of classification.
Complete list of omitted features:
unit_state, _trusted_judgments, _last_judgment_at, fav_number, gender_gold, profile_yn_gold, profileimage, tweet_coord, tweet_created, tweet_location, user_timezone

Corrplot visualization:
The **corrplot** package is a graphical display of a correlation matrix, confidence interval



This shows the varying level of correlation of between the different numerical features. Since the high level of correlation was along the diagonal, I decided to drop these features.
Since I was not doing location analysis, I omitted features such as tweet_location, user_timezone, tweet_coord.

Visualize most frequently used words by all genders:
I combined the features Text and Profile Description. Using the NLTK library I was able to separate the stop words from them. I removed words that were not part of the English language. Also I dropped the rows, whose tweets were not in English. Consider the following pipeline:



The visualization of the most frequently used relevant English words (barring stop words) can be done using word clouds:

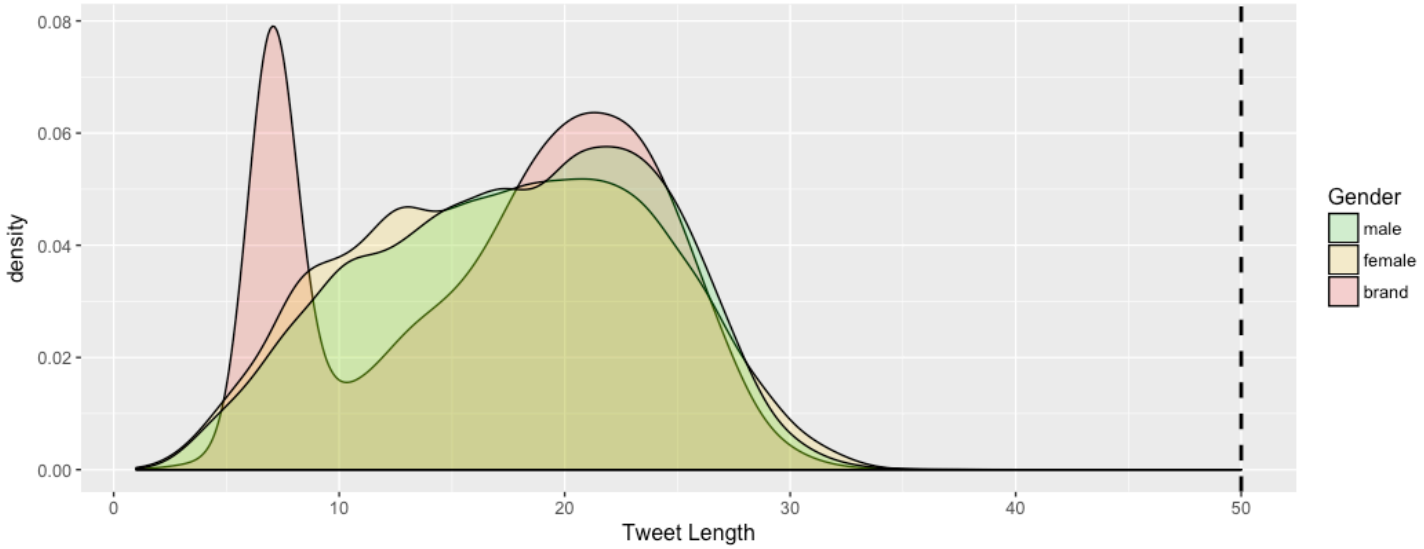
Male word cloud

Female word cloud:

Brand word cloud:

Inference: If we see the male and female word clouds, most of the frequently used words are the same. So we can very well be prepared that our algorithms are going to have a tough time classifying male from female and vice versa if we go ahead and use text and description as our features (just based on frequncy). On the other hand, they will have an easier time classifying brands since the brand word cloud has some distinct words. So in order to use text and description as our features, we will need to add some more modifications.

Visualizing correlation between tweet length and gender:



Tweet length can be an important feature to consider for classification. But as it turns there is not much difference between the tweet length of male or females. On the other hand, brand tweets are either very short or quite long.

Visualizing correlation between frequently used words and other words:
Basically does a particular gender use certain other words with these frequently used words. If we can find out those "other" words and they turn out to be different or even sentimentally different then they can form a good feature.

Frequent words used my males:

love	like	life	time	new	fan
474	460	341	329	275	274

Female frequent words:

love	like	life	time	day	people
697	586	437	349	348	295

Brand frequent words:

Channel	weather	news	new	follow	best
1211	1155	565	364	278	261

This is not very intuitive. So here I will try to find the highest association these frequent words have with other words for every gender:

Association between frequently used male words and other words:

$love

hate	family	music	never
0.06	0.05	0.05	0.05

$like

feel	stuff
0.10	0.06

$life

live	living	hard
0.14	0.09	0.07

$time

part	full	long	first	next
0.13	0.12	0.08	0.06	0.05

$new

via	day	week	check	video
0.09	0.07	0.07	0.06	0.06

$fan

sports	husband	big	dad	football	enthusiast	also	father	lover	music	hard
0.18	0.11	0.08	0.08	0.08	0.07	0.06	0.06	0.06	0.06	0.05

Association between frequently used female words and other female words:

$love
 music hate live always family
  0.08 0.07 0.07 0.06 0.06

$like
  feel look getting
  0.10 0.07 0.05

$life
living live make
  0.11 0.07 0.05

$time
first
 0.08

$day
 every via happy new hope thank mind better
  0.15 0.12 0.09 0.09 0.08 0.07 0.06 0.05

$people
  hate something think
  0.06 0.06 0.05

Association between frequently used brand words and other brand words:

$channel
 continuous price subscribe
  0.27 0.24 0.18

$weather
  nan continuous price
 0.65 0.28 0.25

$news
  latest across technology information daily sports world official tech city
  0.30 0.12 0.11 0.10 0.09 0.09 0.08 0.07 0.07 0.06
  link read top
  0.06 0.06 0.06

$new
video
 0.06

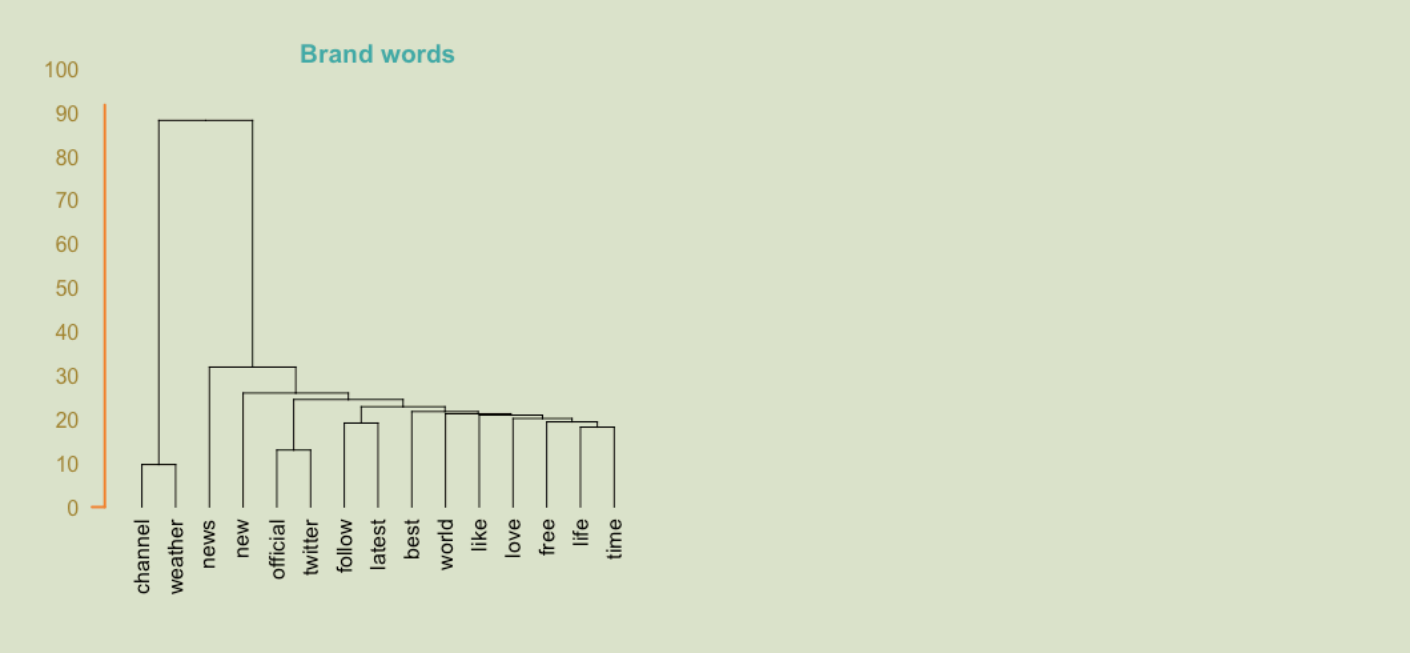
$follow
 latest tweet job account help click need team thanks back official see twitter
  0.21 0.21 0.20 0.17 0.15 0.12 0.12 0.10 0.10 0.09 0.09 0.09 0.09
  find link want
  0.06 0.06 0.06

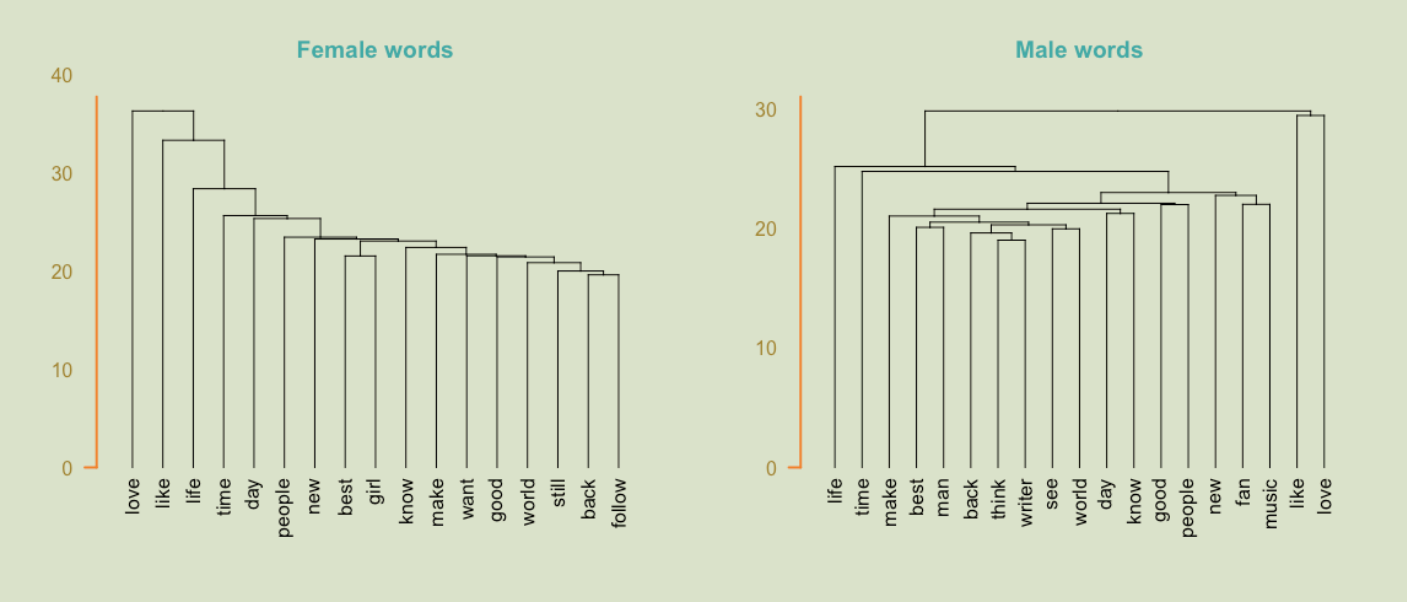
$best

service	community	media	network	radio	live	site	city
0.10	0.09	0.09	0.09	0.09	0.08	0.08	0.07

Inference:
Again we see there is not much to separate between the two genders. The usage of words is quite similar as well as the helper words. Though we infer that men usually talk more about sports and novel things and women talk more about people which is interesting but not very intuitive. Brand shows good results though. This phenomenon will also be highlighted when we test our trained model.

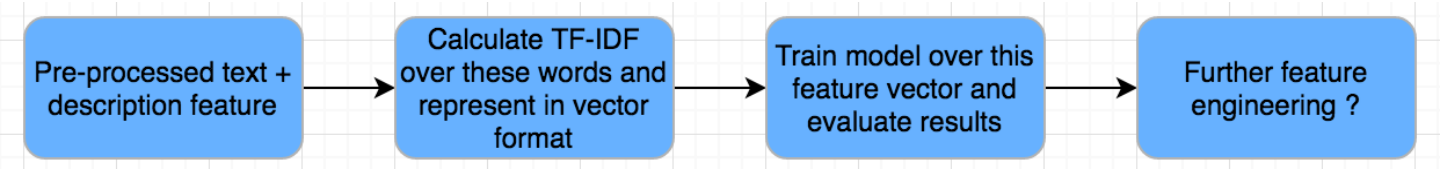
Visualizing association between the frequently used words and other words for each gender:
This gives us a nice hierarchical clustering between the different words. Thus weather and channel are grouped together and closely associated with news.



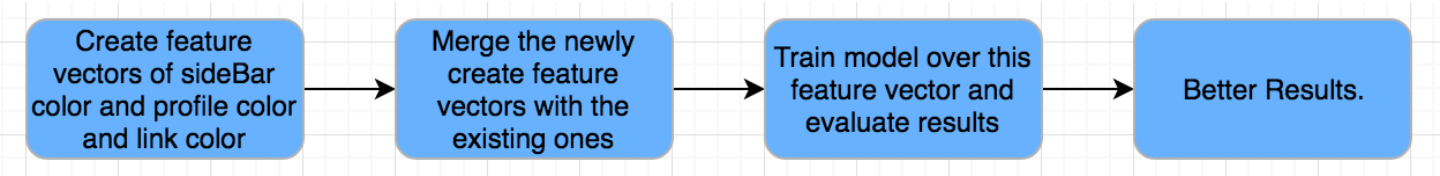


Thus we see a completely different dendogram even though many of the words used by both the genders are quite similar. So the sentiment behind the words can be an important factor to take into account during the classification. Along with sentiment, the grouping of the words can be a really good feature. Even though the frequent words might be similar, their grouping (with other words) varies.

Feature Engineering and Representation:
Before we get on with the evaluation and performance of various algorithms, there are a couple of things consider. Which all features do we need to consider? How can we represent these features in their best form to get maximum accuracy? How do we need add additional features to this mix?



After performing the above steps, all the algorithms below gave results close to 58% (except Random Forests) which is Crowdflower's (Sponsors of the dataset) baseline accuracy. In order to achieve better results I thought of merging two more features namely sidebar color and profile link color.



After this step, I got close to 61% accuracy with Naïve Bayes, Logistic Regression and Linear SVM.

Evaluation of different algorithms: In the below evaluation, I have divided my dataset into 2 parts train and test in the ratio of (80: 20). I have used label encoding to encode the class labels. **So basically 0 corresponds to brands, 1 corresponds to female and 2 corresponds to male (remains consistent for all the algorithms).** I have tried to evaluate the performance of the algorithms using ROC curves and with the help of recall, precision and f-1 score. Also I have presented confusion matrices since accuracy sometimes can be misleading.

Technologies used:

1. NLTK: For cleaning the dataset from stop words, non-English tweets, proper nouns.
2. Pandas framework: For its sheer flexibility and data manipulation and storing the data frame in pickle
3. Scikit learn: For TF-IDF and Count Vectorization, so that I get the most important words with respect to a particular tweet as well as the entire corpus. Also, for the wide array of algorithms, parameter settings, metrics, data set manipulation techniques available.
4. Matplot lib: for plotting and visualizing

1. **Naïve Bayes**: This is one of the best classification algorithms for text analysis and is widely used for spam detection. And sure enough, Naïve Bayes did not disappoint. For K = 25 (for k-fold cross validation). (Program generated output)

('Accuracy without k-cross validation', 59.246392303580975)
('Accuracy with cross validation', 60.45601085421076)

Area under the curve for brands 0.83398685493
Area under the curve for female 0.768461106559
Area under the curve for male 0.734388823805

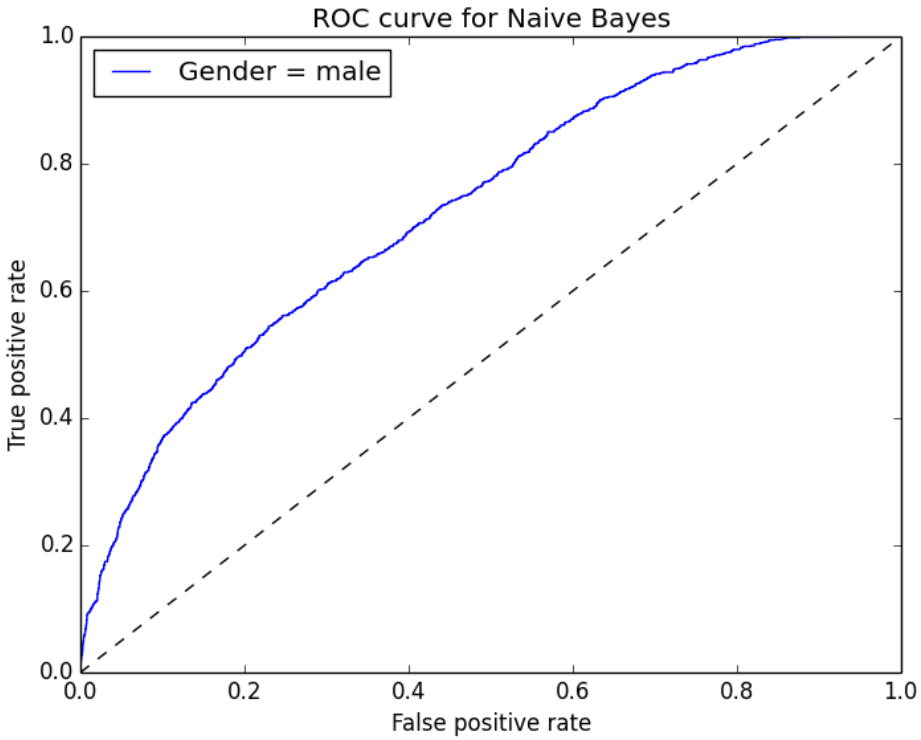
Classification report:

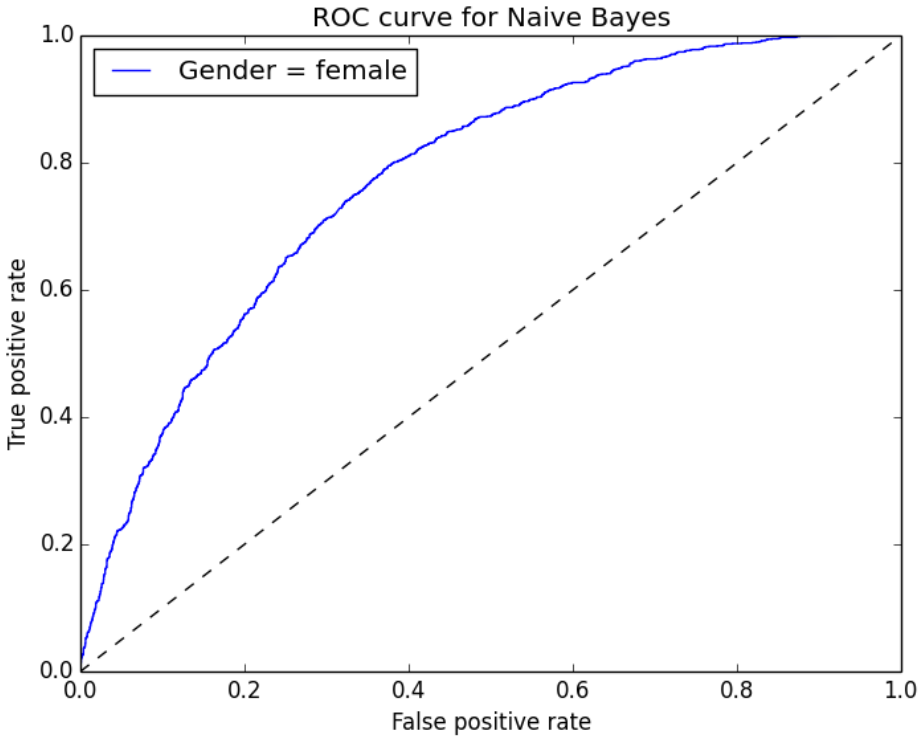
	precision	recall	f1-score	support
0	0.72	0.62	0.66	1190
1	0.56	0.67	0.61	1333
2	0.53	0.49	0.51	1219
avg / total	0.60	0.59	0.59	3742

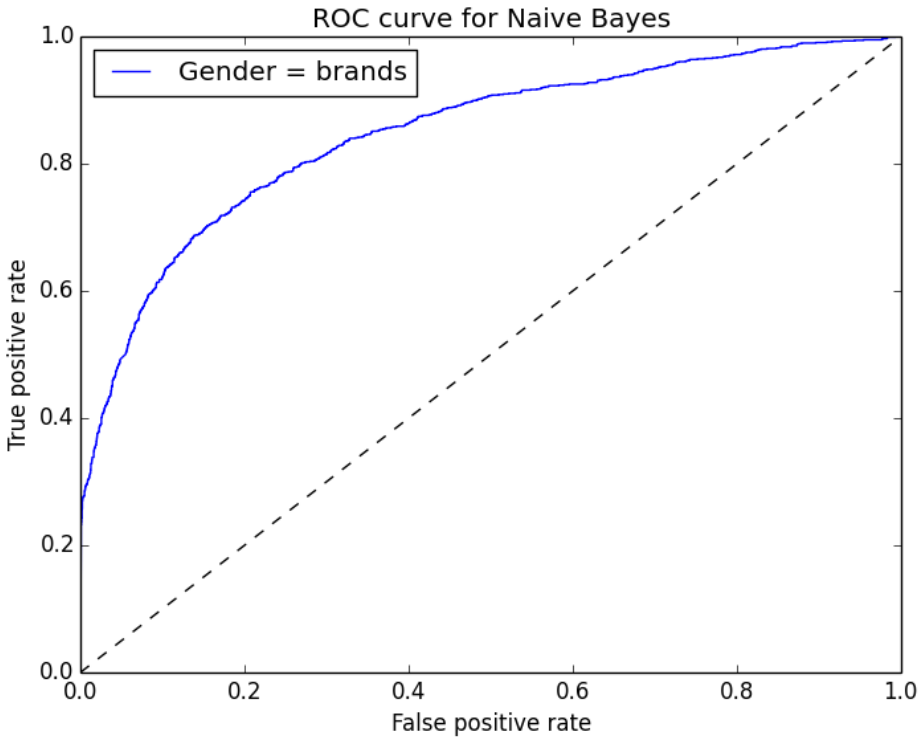
Confusion matrix:

Predicted ->	0	1	2
Actual 0	[[734	231	225]
1	[137	887	309]
2	[149	474	596]]

ROC curves:


ROC curve for Naive Bayes
Gender = male
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0


ROC curve for Naive Bayes
Gender = female
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0



Inference and analysis: We observed in EDA that there is no clear distinction between features of the male and female gender and this can be very well being seen through their ROC curves. The high area under the curve is obtained for brand and the lowest for the male gender.

Failure: Even through Naïve Bayes is really good in text classification and generalization, it falls short when it comes across features that are occurring in both the classes (male and female). Since it does not associate weights and depends on the general count, we see LR (Logistic Regression) performing a tad bit better. Also it performs badly against unseen data. Also I feel due to Naïve assumption (conditional independence of the features), the results are having an error since I feel there is certain amount of dependence between the features I have chosen. LR does take that into account and hence should perform better.

2. **Logistic Regression:** As expected Logistic Regression does provide better results than Naïve Bayes with almost 3% accuracy jump. Since it provides weights, it performs better for samples that are unseen.

('Accuracy without k-fold cross validation', 61.945483698556927)
('Accuracy with k-fold cross validation', 60.869531716740831)

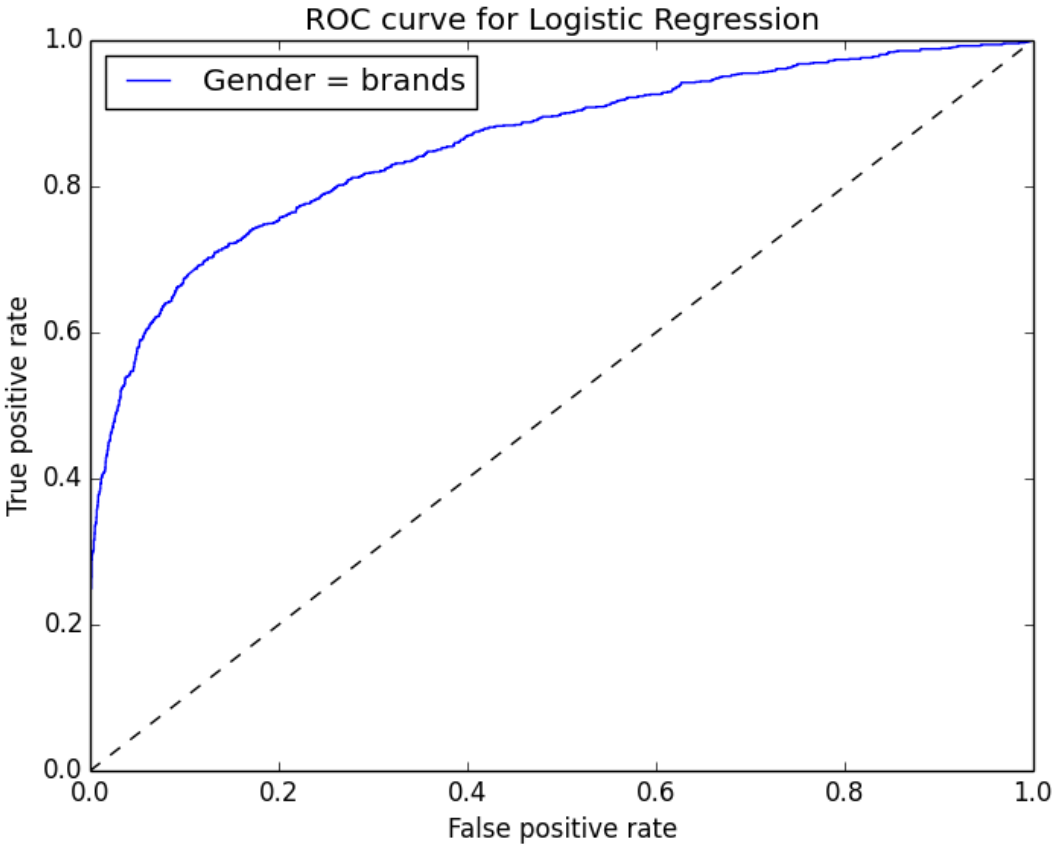
('Area under the curve for brands ', 0.8575557658761086)
('Area under the curve for female ', 0.78175875716585574)
('Area under the curve for male ', 0.72821839939678368)

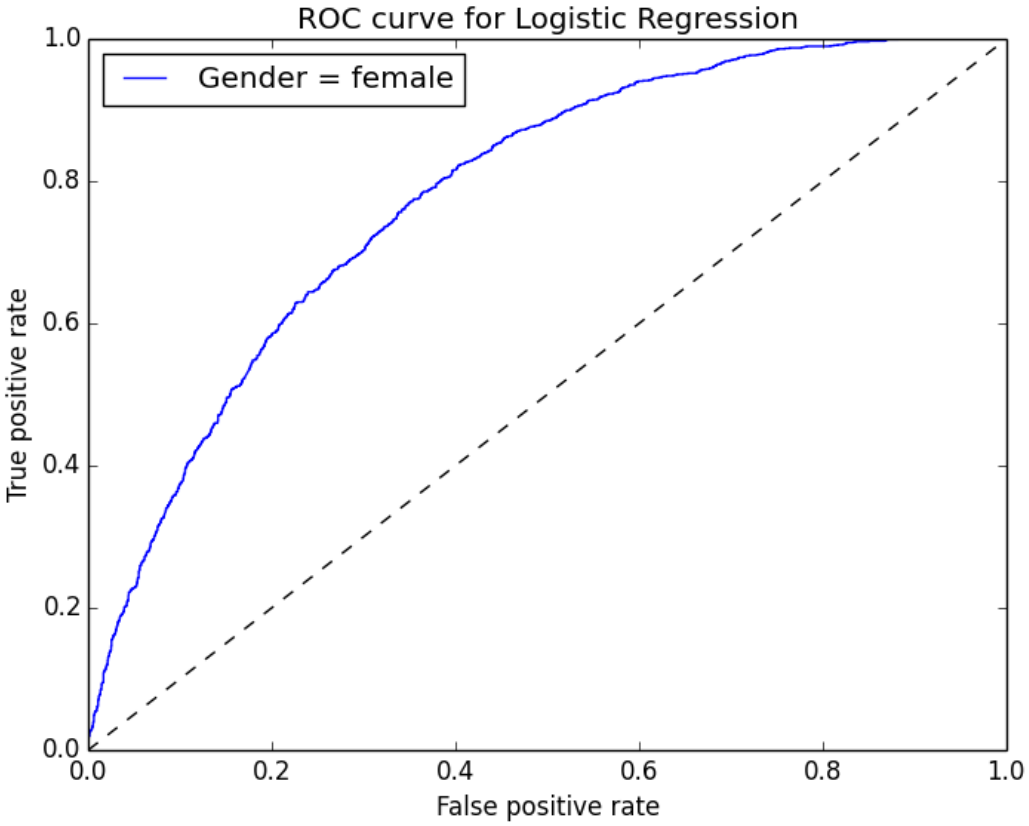
Classification report:

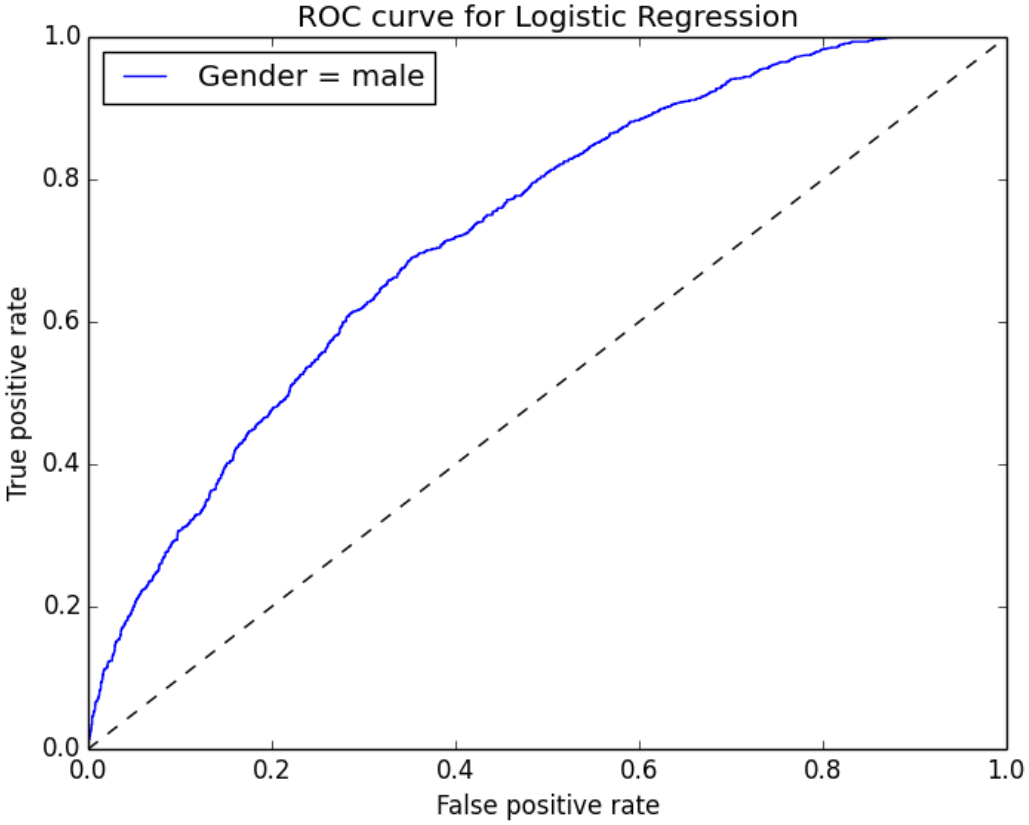
	precision	recall	f1-score	support
0	0.76	0.66	0.71	1176
1	0.59	0.68	0.63	1346
2	0.53	0.52	0.52	1220
avg / total	0.63	0.62	0.62	3742

Confusion matrix:

Predicted ->	0	1	2
Actual 0	[[778	180	218]
1	[101	911	334]
2	[142	449	629]]




ROC curve for Logistic Regression
Gender = female
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0


ROC curve for Logistic Regression
Gender = male
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

Inference, analysis and failure: Our hypothesis does hold that LR performs better than Naïve Bayes but still the performance is not up to the mark. Again, I feel the features at disposal weren't up to the mark. Through this exceeds the baseline accuracy by quite some margin and classifies brands pretty well, it struggles when it comes overlapping features belonging to both male and female class. Though the area under the curve has improved, we see ROC curves that are quite similar to the ones of Naïve Bayes.

Random Forest (RF): In order to satisfy my curiosity, I went on and tested with Random Forest having 100 trees has estimates. I did not have much expectation since RF fails miserably when it there isn't a clear feature distinction. When I say there is no clear feature distinction, I mean the features which we used (Text, Description, Side bar color, profile link color) are quite common, between both the genders (male and female).

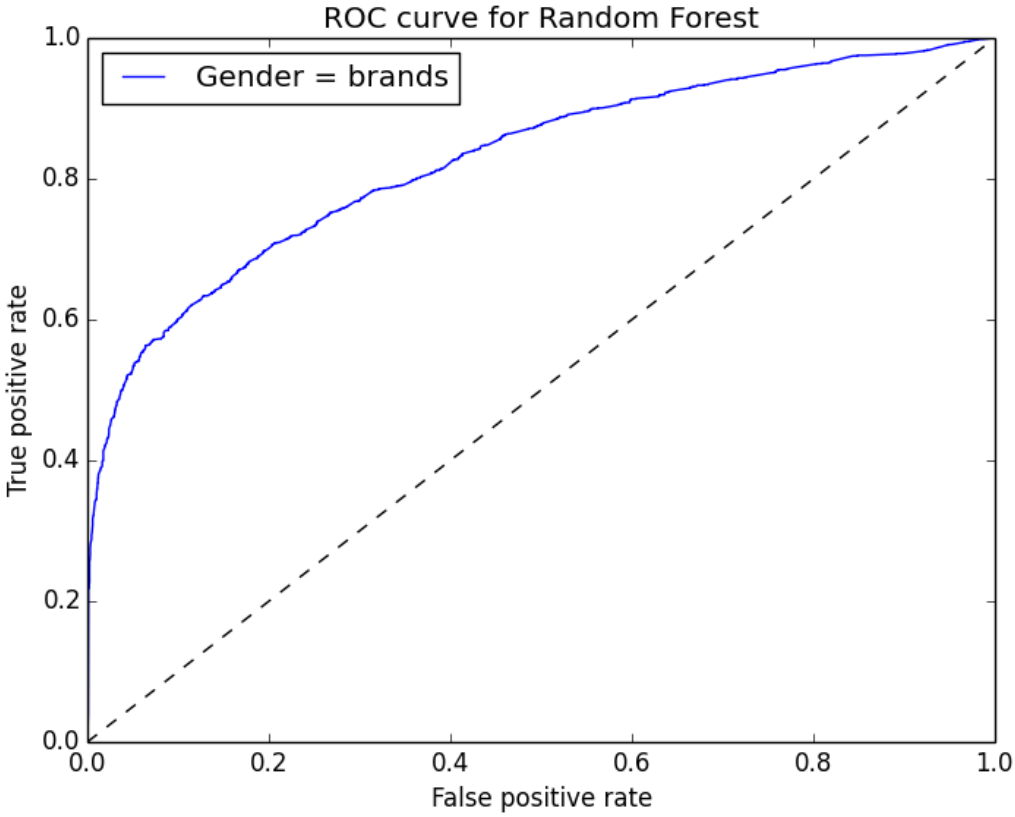
('Accuracy without k-cross validation', 59.406734366648848)
('Accuracy with cross validation', 58.204090891229875)

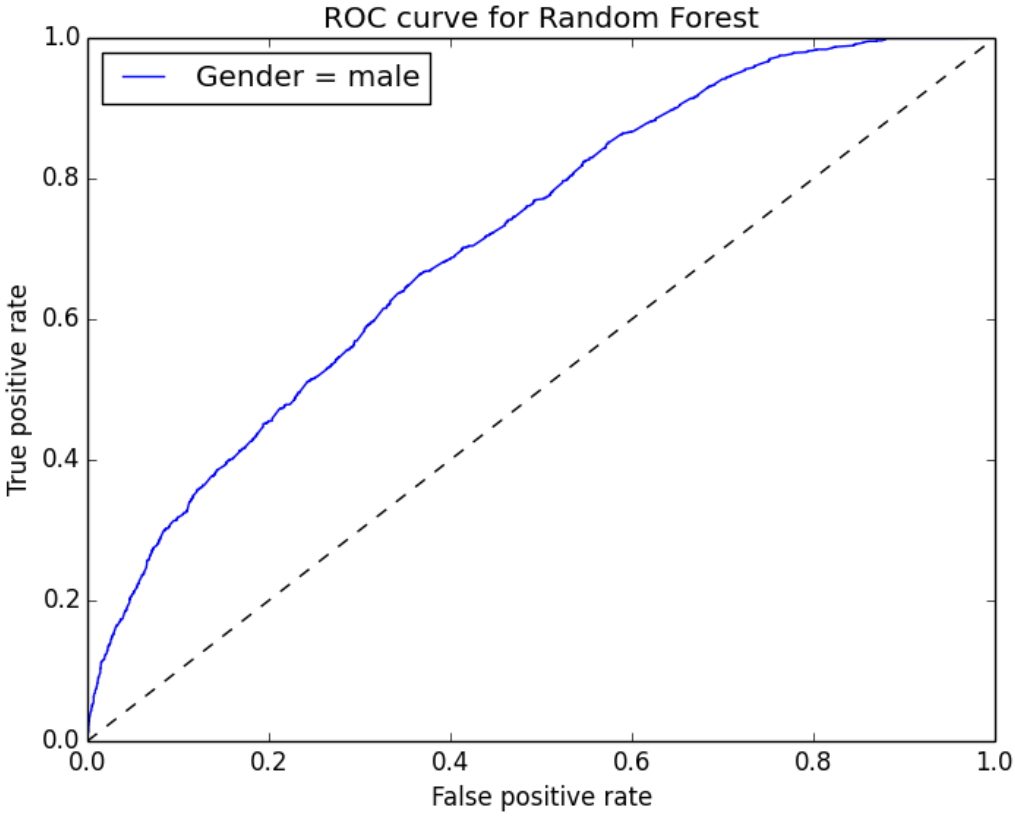
('Area under the curve for brands ', 0.83469548419534489)
('Area under the curve for female ', 0.76497693844107784)
('Area under the curve for male ', 0.71158045887654553)

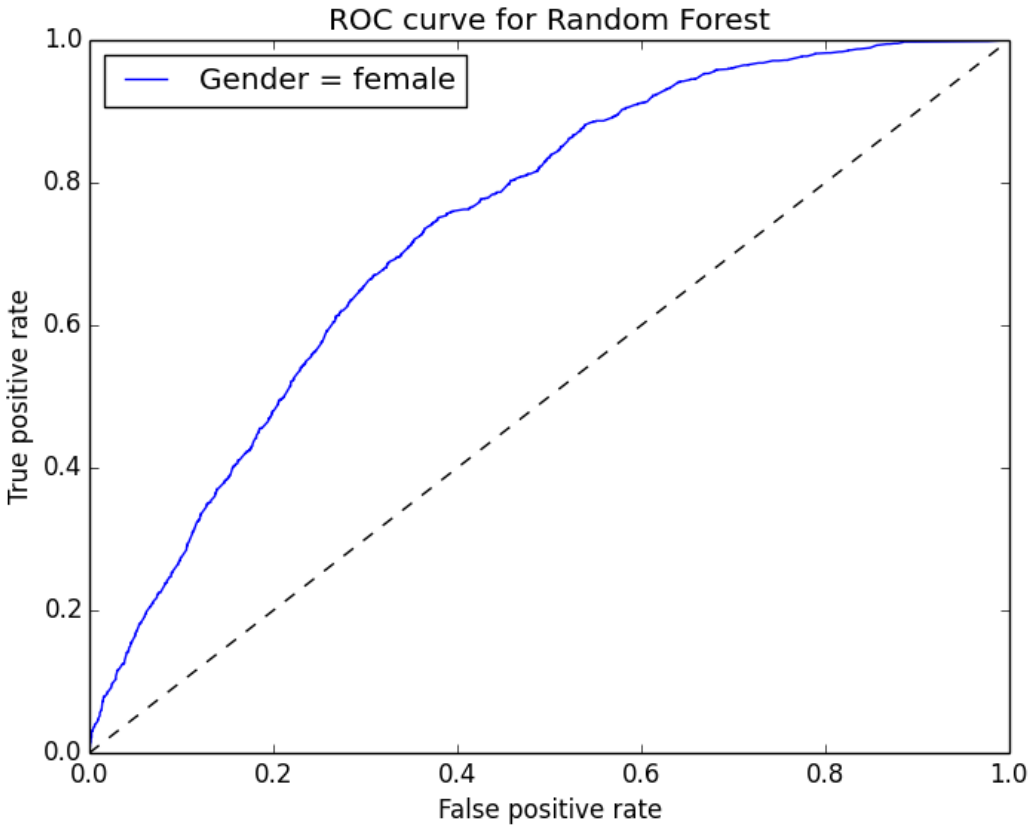
Classification report:

	precision	recall	f1-score	support
0	0.74	0.63	0.68	1180
1	0.54	0.71	0.62	1334
2	0.53	0.43	0.48	1228
avg / total	0.60	0.59	0.59	3742

Predicted ->	0	1	2
Actual 0	[[744	246	190]
1	[110	948	276]
2	[147	550	531]]


ROC curve for Random Forest
Gender = brands
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0


ROC curve for Random Forest
Gender = male
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0



Inference, analysis and failure: As predicted the accuracy falls. Random Forest (RF) is not a good solution, its hypothesis space is a rectangle and we clearly see here that the data points are quite over lapping. Even though RF does not expect something that is linearly separable, its decision boundaries fail if there is not clear feature split. Where it finds distinct features, it performs reasonably well (see brands ROC).
Hence we find absolutely linear classifiers performing well.

Final thoughts and challenges:

1. I feel that the features at disposal weren't the best ones. If we observe that there was just one random tweet per person and the profile description. If we are to predict something like gender from that we need more information. It would have helped if we had more no. of tweets per user.
2. Also I do not blame the dataset for not having absolutely distinct features. There is not much difference between the two genders in the manner in which they present themselves on the social media.

Evaluation of objectives: Kindly refer to the objectives mentioned at the start, as I go through them

1. As we saw that by adding more important and relevant features we got an increase in the algorithmic performance. That is after adding features such as sidebar color, profile color and profile description to the text feature vector we were able to classify better.
2. We got an overview through different visualizations regarding the most frequently used terms by all genders, correlated terms, relation between tweet length and gender and association between the frequently used words and other words per gender.
3. I feel predicting gender just based text of tweets is not an easy task. There is a need to add more features to get better performance.
4. There are no such words that "strongly" predict male or female gender. Though there is a certain emotion behind it during their usage that differs across the genders. Also we find (through dendograms) that the words used along with "most frequent" words can be a useful criterion. Hence I feel a tri-gram or bi-gram model would have performed better. (Future scope)
5. Profile and link colors did prove out to be useful factors and helped in boosting the accuracy.

References:

1. http://scikit-learn.org/
2. www.stackoverflow.com
3. www.kaggle.com
4. www.quora.com
5. http://www.aclweb.org/anthology/C14-1184
6. http://www.aclweb.org/anthology/D11-1120

Acknowledgement:

Special thanks to Prof. Sriraam Natarajan and the AIs for making this a wonderful course.