1 Data preparation

The data for this work was prepared by directly downloading tweets from twitter website. Based on a preliminary scan of makeup related tweets, we prepared a list of hashtags and words that could potentially help us find the twitter users we needed for our study. However, given the nature of the study and the search timings, the collection built using this search resulted in being predominantly spam. Subsequently, we built a list of major cosmetics companies and began collecting their followers. Collecting the followers and growing the company list was carried out as as an iterative process: the followers collected in the first round were ranked in the order of their probability of their being active on cosmetics company sites. The companies followed by active users were added to our prior list of companies (in general, 5 in one loop), their followers were downloaded again and subsequently used to gather names of more makeup brands. Using this (somewhat) focused crawling, we built a list of more than 100,000 users. We remark here that two big limiting factors in our search were (i) we did not have a good language translator, so we had to limit our search to users using English only, and (ii) we had to ignore users who had 'protected' profiles because their tweets are not downloadable.

Once the initial set of users was ready, we downloaded the timelines of each user. Twitter allows downloading up to 3,200 tweets for any given user. Using the set of users we had, we were able to build a set of more than one million tweets. Next, we went back to the tweets we had orignally downloaded (the collection mentioned in the beginning of this section, where the tweets were predominantly spam), extracted spam tweets, combined those with equally many good tweets, confirmed the correctness of this collection, and used this set to prepare the training data for a spam classifier. Since the vocabularies of spam tweets vs non-spam tweets have strong dependence (exclusive OR, for all practical purposes), we used the *Naive Bayes* classfier to model the spam classifier. The accuracy of this classifier was tested on a set of new spam tweets that we collected again from twitter (this time actually using spam keywords). Our classifier was able to classify the test tweets with 99% accuracy. The correlation matrix for test experiments is shown in table 1.

Apart from the spam tweets filter, another too we needed was a strategy to distinguish the followers of our study-case users. To this end, we noticed that, in general, the $followers\_count$ of twitter users typically obeys the pattern 'less than 2500 for most single users (and often very small businesses such as beauty salons), in between 2500 to 5 million followers for most businesses, and more that 5 million followers for celebrities'. While this rule certainly had outliers (especially, for the single user case), for the scope of this study we chose to adhere by this rule, treating very small businesses as single users at current stage. We chose to use this as a preliminary criteria for evaluating the followers of any user of interest (a company itself or a single user) for the rest of our study. In addition, we also prepared a list of same number of non-cosmetics companies (news channels, sport related twitter communities, newspapers religious communities) and collected their timelines. These varied tweets were combined with the tweets from cosmetics companies (ensuring fair representations) to prepare the training data for a second classifier: one that would classify cosmetics-company-tweets vs non-cosmetics-company-tweets and can be extended to classify cosmetics based businesses from noncosmetics based aforementioned large public popular entities. However, we cound not eventually settle on a classifier that would give satisfactory results for us. Wherever needed, therefore, we had to adhere to the first two criterions for segregating businesses appropriately.

Next, we started evaluating the collection of users with less than 2500 followers (i.e., the ‘individual’ users). Our first step was to discard any spam tweets in the entire collection of tweets. We used our spam classifier to this end. We remark that the spam tweets in our ‘focus-crawled’ tweets collection was very minimal. Furthermore, we also ruled out any users whose timeline was less than a year.

From this filtered dataset, we collected the list of owners of the constituent tweets. This being still a very large set (approximately 90,000 users), we still needed to narrow down our search space for training data set candidates. We used topic modeling to this end. Specifically, *for each individual user* in this list, we segmented his tweets into constituent topics using latent dirichlet allocation (LDA) based topic modeling. In addition to the twitter imposed limit of 3,200 tweets, another aspect of the timelines was that different users had different number of tweets. Due to this, blind LDA on these timelines fails. Due to this, we chose to divide the user's timeline in a flexible manner: we required that the timeline be divided into K categories such that $\frac{N}{K} = 10$. This strategy allowed fairer distribution of categories and yielded more accurate results. For each category, we took 6 representative words. (For ease of presentation, let's call these representative words as the basis vectors for the corresponding category). We considered the intersection of the basis for each category with the set of our cosmetics keywords. The resulting set for a given category was a preliminary indicator of a user's interest in cosmetics. The bigger the intersection, and more the categories yielding non-trivial intersections, the higher the interest. In this way, we set an initial profile for our entire set of single users: those with no categories as cosmetics were perhaps completely not cosmetics oriented, those with approximately half the categories as cosmetics were probably fairly cosmetics oriented, and those with majority of the categories as cosmetics were perhaps either cosmetics obsessed or small beauty-businesses. Table 1 displays some topics that the algorithm learnt for two users belonging to different bands of the spectrum.

beauty dependent	beauty hacks moments false french girl red lipstick makeup teeth facial collection colorlicious bloom amp http win chance #12daysofbenefit cosmetic jewelry products skin tips absolutely issues #beauty glowing hair dry wrong jawdroppingly professional damas man enjoy wowza body learn week complexion palette holiday love time #loveison #elfholiday gift wha
partially interested	#makeupartist #makeup #makeupluv03 #makeupblogger blogs team #pinaymakeup app real chat download #instagramers! messaging hizaa love nail art suggestions beauty skin awake today cc cream mini natural mascara love watching love good day #viggle makes lmfao day hope daughter happy prom care followers night sh friends great ladies playing weekend

Table 1: Example of LDA of 2 users: The first user had 8 categories in all and the second user had 23 categories in all. For space considerations, we have displayed only 7 categories of each user.

After LDA, we started evaluating our users against the features provided by the psychiatry chart provided to us by domain experts. The chart has seven different check-list attributes. We chose each of these features as a feature vector for our final data matrix (that wills be fed to a classifier). Our strategy to fill out each feature vector was as follows:

1. Tolerance: We graded the tolerance of the user towards cosmetics on a scale of 0 to 4 depending on how far the user had gone with his interest in cosmetics. A user with no keyword occurance was graded at 0, a user with cosmetic surgery keyword occurances was graded at 4.

2. Withdrawal symptoms: This one was smore difficult to track so we left it for later in the interest of time.

3. Consumption: We judged this based on the user's declarations of his/her shopping of cosmetics.

4. Regret: This was a direct keyword based search on the user's timeline.

5. Compromise on essential activities: We looked for keywords that indicated whether the user spent money/time on cosmetics at a juncture when he/she was supposed to use it for other purposes (eg., food, studies, shopping necessaties for self/family etc.)

6. Time/resources spent: We noted the rate of increase in the user's cosmetics related activity as the average rate of change as reflected in his cosmetics-tweeting frequency (including retweets), number and frequency of shopping related tweets, and any tweets that contained '$' signs.

7. Sorry-not-Sorry: This, again, was a direct keywords based search carried out using the relevant keywords.

   Evaluating (4) and (7) above, required scanning not only the current tweet but also some (we considered 5) tweets prior to it. An example of what would qualify as a regret tweet would be:

2 Results and discussion

The results of our LDA analysis were encouraging. They helped immensely in improving the productivity of our search for candidate twitter users. Taking the entries in our preliminary data matrix as indicators, we were able to prioritize users. We essentially went by the number of nonzero columns in our preliminary data matrix. Users with higher number of nonzero columns were potentially more dependent while users with two or less nonzero columns were expected to be mostly not dependent. This determined the order in which we searched for potential candidate users on twitter. The final data matrix that we prepared consisted of about 1500 users, of which about 50 displayed obvious dependence on cosmetics, approximately 500 were picked to be completely independent and the rest showed varying levels of dependence. We put this data in three classfiers: the Gaussian Naive Bayes, the multinomial Naive Bayes, and the Bernoulli Naive Bayes. We used the python inbuilt implementations of these classfiers with $k = 2$ folds. The precision, recall and f1 rates for the resulting classfiers are displayed in

Due to our constraints on tweets availability and such other factirs, our current training data is still small to provide good confidence decisions. Our plan was to input this matrix into the multinomial Bayes Classier and the Gaussian Classifier and evaluate the resulting model against a set of about 20-31 test users we had hand picked in the beginning of the project.

The main codes for this project were written in python. Due to the magnitude of data and the nature of data processing involved in this study, all the twitterdata we downloaded was stored and accessed via postgres database. The overall queries were combinations of python and sql.