ADVANCED PROGRAMMING
Project 2b Report

CSE201 MONSOON'15
VARUN JAIN, AYUSH AGARWAL
2014170, 2014029

We started out with registering the app on Facebook and retrieving data from our Facebook profile to perform the prediction analysis of likes on it.

Considering the case that we took a recent post on our timeline. On this post we have developed three algorithms to predict the possible number of likes and the people who will like this post by performing the analysis of likes on previous posts on our timeline. We will be looking at the results from all three individually and also attempt to make some final and definite conclusions based on these results.

- ## *algoMach1*

  This is where the real stuff happens. After looking at all the posts and taking a certain upper limit of them (about 80% of the posts) we apply the first step of our algorithm to analyze these posts. We start with creating a hashmap which maps the names of our friends liking our posts to the posts they have liked. Then we extract all the words of these posts and store them in an arraylist. This will serve the purpose of aggregating all the words (other than the common dictionary words), that have been liked by our peers.

  Now we look at the new post that we have put up, make another arraylist and store all the words of this post into the new arraylist.

  The next step involves us looping through all the words of the new arraylist and check whether these words also occur in the old arraylist. Correspondingly, we keep incrementing the number of matches that occur in the new post with all the previous posts.

  Finally we have a quantity with us which denotes the matches. We divide this with the total size of the new post, which gives us the probability of this post being liked by the particular person for which the algorithm is being run at that instant.

  Keeping a minimum threshold for the probability (given by the 'Optimism Level'), we can say that the person will like this post. Immediately this leads to an increment in the count which ultimately shows the predicted number of people liking the post. The threshold is again taken with the help of the slider from the user.

- *algoMach2*

  The second algorithm is a more direct one which takes each liker individually, checks that how many of the person's posts he/she has previously liked and correspondingly sets the probability of liking. If this probability is greater than a certain percentage (the optimism level), count is incremented and the person is considered to surely like the post.

- *algoMach3:*

  The third and the final algorithm is the conventional root of mean of squares of the number of likes on each post on the person's timeline. This is the base algorithm whose results are independent of the post and the people liking it. The result returned is always a fixed quantity in this case.

- *Invariants*

  Certain invariants/ thresholds have been applied to ensure accuracy of results

  1. The number of likes on a post dated today are independent of the likes on a post 3 years back.
  2. Certain weights have been assigned each to the result from algoMach1 and algoMach3 to arrive at a definite like count.
  3. The majority of likes for most of the people appear not on text heavy posts, but on photos. Hence the likes predicted for a photo (especially a profile picture) will always be significant.

We obtained the following insights after the analysis with various algorithms:

1. algoMach1 is quite accurate. According to the optimism index, it looks at the number of words matching in the post and accordingly increments the count.
2. algoMach2 gives really small values: In this case even if your optimism level is a sky high 90%, which requires only 10% of the posts to be liked by the people, we observe that we get very small values for this.
3. algoMach3, which computes the rms, keeps giving out a constant value which is a good measure (can be used as a base against which algoMach1 can be evaluated). It is generally quite close to the actual answer.