MengProject

Arjun Jauhari

March 2016

1 Data

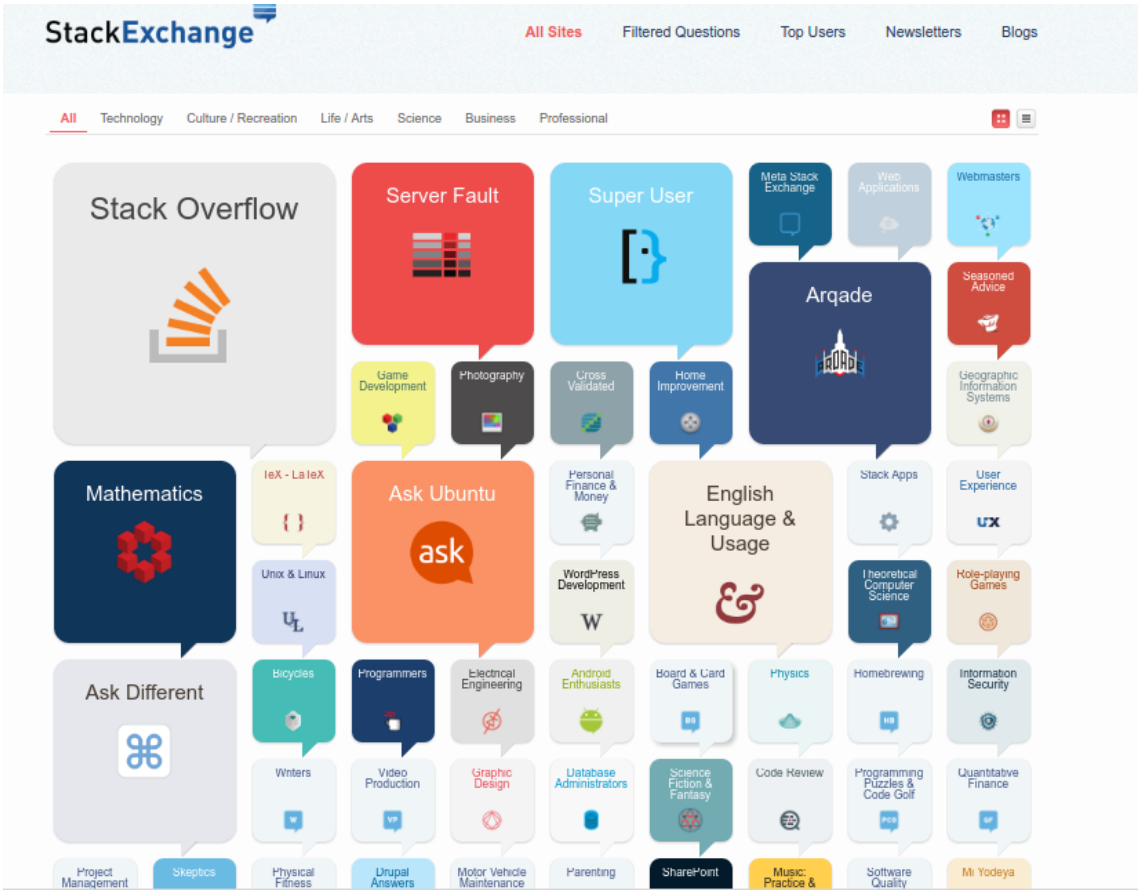

Figure 1: Sample Graph

Data is structured into following files

- Badges.xml
- **Comments.xml**
- PostHistory.xml
- PostLinks.xml

- **Posts.xml**
- Tags.xml
- **Users.xml**
- Votes.xml

A typical row in Posts.xml file looks like

$< row$**Id** = "1"**PostTypeId** = "1"**CreationDate** = "$2015-02-03T16:40:26.487$"**Score** = "22"**ViewCount** = "307"**Body** = "$SampleQuestion$"**OwnerUserId** = "2"**LastEditorUserId** = "2"**LastEditDate** = "$2015-02-03T17:51:07.583$"**LastActivityDate** = "$2015-02-03T21:05:27.990$"**Title** = "$SampleTitle$"**Tags** = "$line-numbers$"**AnswerCount** = "2"**CommentCount** = "0"**FavoriteCount** = "3"$/>$

2 Extracting Attributes

Wrote shell scripts to extract out the attributes from the above mentioned data files. Attributes used for the project were:

2.1 From Posts.xml file

Id
PostTypeId
OwnerUserId
ParentId(onlyforans)
Score
AcceptedAnswerId
CreationDate
AnswerCount

2.2 From Users.xml file

Id
DisplayName
Reputation
UpVotes
DownVotes

2.3 From Votes.xml file

Id
PostId
VoteTypeId
CreationDate

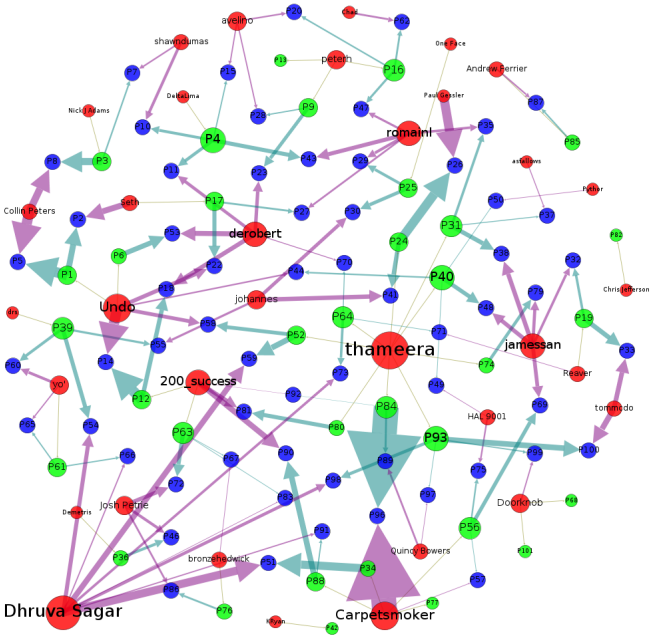

Figure 2: Sample Graph

3 Generating Gephi graphs

To visualise this model I generated graphs using Gephi, figure 1 above shows one graph on a very small subset of data. Legend for the graph-

- Red Nodes: User
- Green Nodes: Questions
- Blue Nodes: Answers
- Green Edge: User2Question(u2q)
- Blue Edge: User2Answer(u2b)
- Grey Edge: Question2Answer(q2a)

The edges connect each user to their post. Post can be a question(green edge) or an answer(blue edge). Also, each question is connected to its answers through grey edge.

Important details about the graph -

- Node size represents, how active is that particular user. For example, user with name 'thameera' is the one of the most active user followed by user 'Dhruv Sagar'.
- The thickness of these q2a edges tells the quality of an answer relative to all the other answers for same parent question. For example, user 'Carpet Smoker' has give one of the best answer.

4 Modelling as System of Linear Equations

4.1 Subscripts

- i is subscript for question.
- j is subscript for answer.
- k is subscript for user.

4.2 Notations

1. u_k: Quality measure of the kth user.
2. q_i: Quality measure of the ith question.
3. va_{ij}: Normalized votes corresponding to jth answer of ith question.
   Calculated as: $va_{ij} = \frac{|sa_{ij}|}{\sum_j |sa_{ij}|}$
   where sa_{ij} is the actual votes(score) read from data dump.
4. a_{ijk}: Quality measure of a_{ij}th answer given by the kth user.
5. f_{acc}^{ij}: Boolean flag telling if this answer was Accepted, read from data dump.
6. r_k: Reputation of the kth user, read from data dump.
7. N_a^i: Number of answer to ith question, read from data dump.
8. vq_i: Number of votes to ith question, read from data dump.

4.3 Equations

Below equations model the relation/dependence between the above defined parameters.
Bold values are known features. All the features were scaled between 0 & 1

1. $a_{ijk} = f_a(u_k, \mathbf{va_{ij}}, \mathbf{f_{acc}^{ij}})$
   $a_{ijk} = 1/3 * u_k + 1/3 * \mathbf{va_{ij}} + 1/3 * \mathbf{f_{acc}^{ij}}$
2. $u_k = f_u(\{a_{ijk}\}_{ij}, \{q_i\}_k, \mathbf{r_k})$, where $\{a_{ijk}\}_{ij}$ is set of all answers by user k
   $u_k = 1/3 * mean\{a_{ijk}\}_{ij} + 1/3 * mean\{q_i\}_k + 1/3 * \mathbf{r_k}$
3. $q_i = f_q(u_k, \mathbf{N_a^i}, \mathbf{vq_i}, \sum_{\mathbf{j}} |\mathbf{sa_{ij}}|)$, where $\sum_j |sa_{ij}|$ is sum of votes for all the answers of ith question
   $q_i = 1/4 * u_k + 1/4 * \mathbf{N_a^i} + 1/4 * \mathbf{vq_i} + 1/4 * \sum_{\mathbf{j}} |\mathbf{sa_{ij}}|$

4.4 Solving this model

All the above 3 set of equations are combined together to form a system of linear equations.
$Ax = B$, where A is the coefficient matrix and B is the vector of all the known quantities. x is the vector of all the unknowns namely : a_{ijk}, u_k, q_i

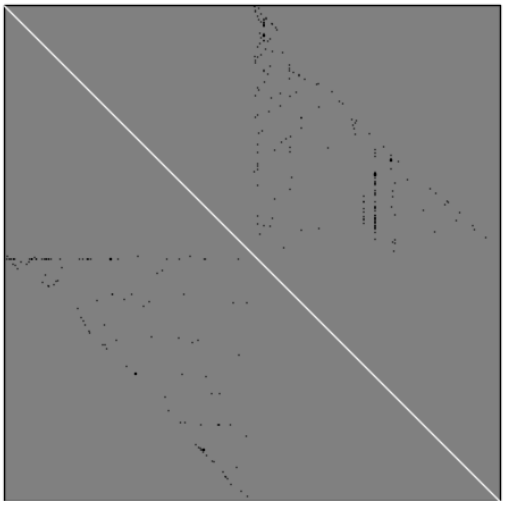

Figure 3: Sample Graph

4.5 Results from this model

Structure of typical A matrix looks like: (white = positive, black=negative, grey = zero)
Precision at K plot, where K is number of Users being considered
Histogram of User quality and question quality

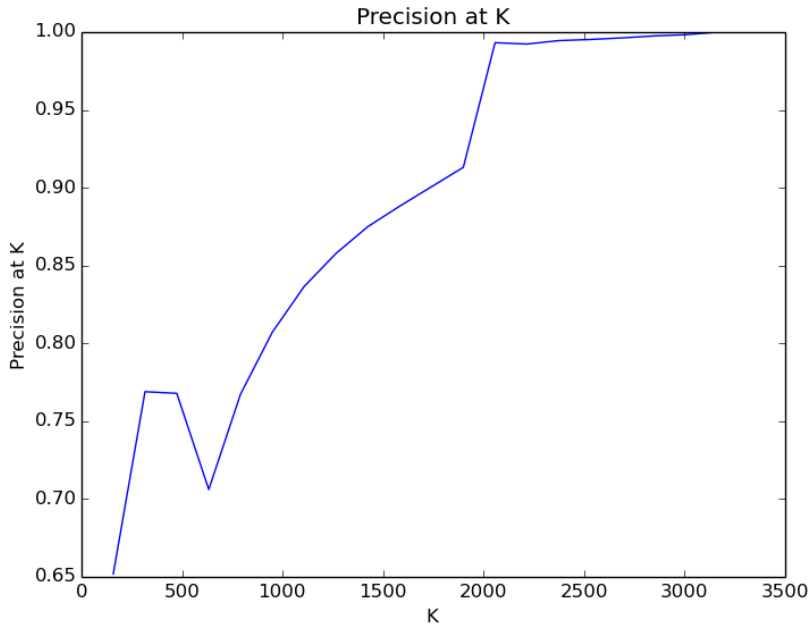

Figure 4: Sample Graph

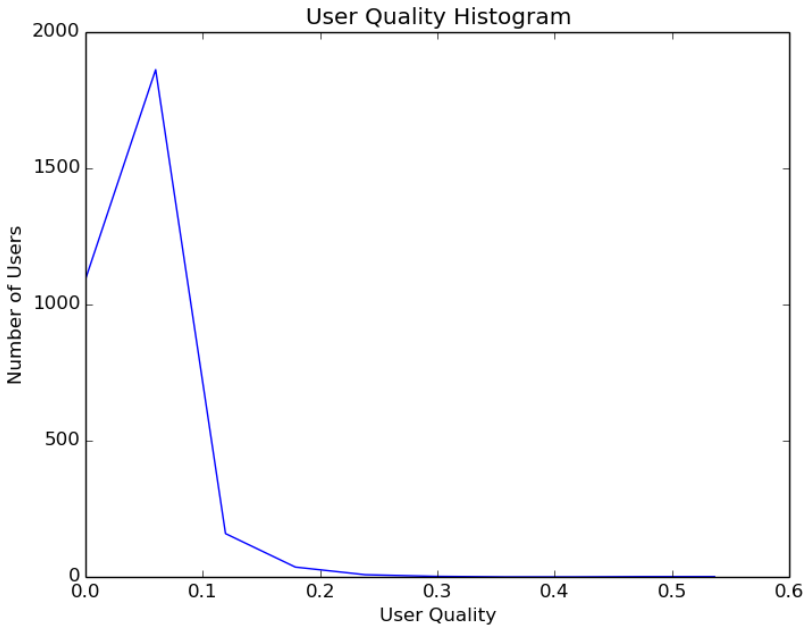

Figure 5: Sample Graph

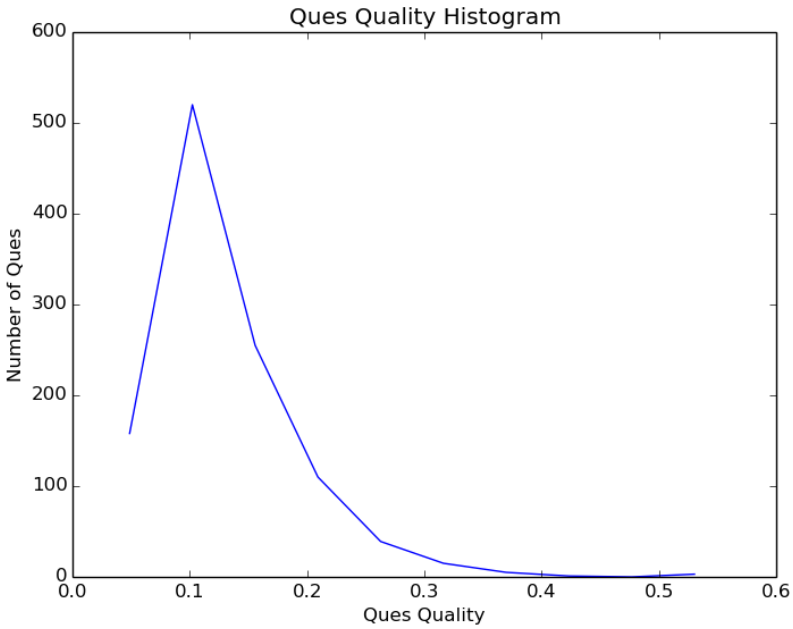

Figure 6: Sample Graph

5 Probabilistic Graphical Model based approach

5.1 Preprocess data

Pre-processed data files to Structure them into useful data structures which can be used in model simulation.
One of the main data structure was a dictionary mapping each question to its click history. For example,
{Q1 : list of click History}.
Further click History was structured as list of list
$[[vote1, [listofanswerspresent]], [vote2, [...]], ...]$
Note: Each click can either be an UpVote or DownVote.

6 Model Simulation

6.1 Graphical Model

Below is the graphical model used. Red node color corresponds to observed variable.

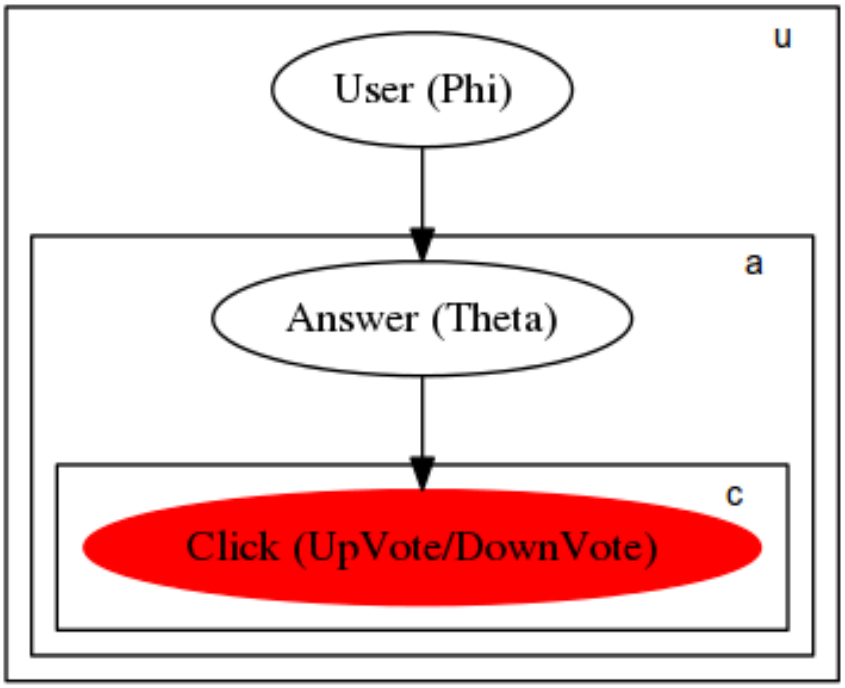

Figure 7: Sample Graph

6.2 Equations

Goal is to find the probability $P(\theta, \phi|clicks)$
And by Bayes rule,

$$P(\theta, \phi|clicks) \propto P(clicks|\theta, \phi) * P(\theta, \phi)$$
$$P(\theta, \phi|clicks) \propto P(clicks|\theta, \phi) * P(\theta|\phi) * P(\theta)$$

Now, we defined the conditional probability as

$$P(click = k|\theta_1, ..., \theta_n) = \frac{exp(\theta_k)}{exp(\theta_1) + ... + exp(\theta_n)}$$

and

$$P(\theta_i|\phi_j) \sim \mathcal{N}(\phi_j, \sigma^2)$$
$$P(\theta_i|\phi_j) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(\frac{-|\theta_i - \phi_j|^2}{2\sigma^2})$$

and

$$P(\phi_j) \sim \mathcal{N}(0, \sigma^2)$$

6.3 Objective Function

The objective function is the product of the 3 terms defined above for every $user(\phi), answer(\theta)$ and $click(k)$.

The goal is to minimize this function and we explored several gradient descent based optimization methods.
L-BFGS : Limited-memory BFGS (L-BFGS or LM-BFGS) is an optimization algorithm in the family of quasi-Newton methods that approximates the BroydenFletcherGoldfarbShanno (BFGS) algorithm using a limited amount of computer memory. It is a popular algorithm for parameter estimation in machine learning.
AdaGrad : AdaGrad (for adaptive gradient algorithm) is a modified stochastic gradient descent with per-parameter learning rate, first published in 2011. Informally, this increases the learning rate for more sparse parameters and decreases the learning rate for less sparse ones. This strategy often improves convergence performance over standard stochastic gradient descent in settings where data is sparse and sparse parameters are more informative.

7 Moving to BIG DATA

Till now all the experiments were based on data set from smaller stack exchange websites like "http://vi.stackexchange.com/" and "http://datascience.stackexchange.com/". In these websites the Number of Users were around 13,000 and number of Posts were around 7,000 and the data set size was $\sim$ 15MB.

Once the model was tested on smaller websites, we moved to our final target i.e. "stackoverflow.com", which has

- Number of Users = 5277830 ($\sim$ 5 million), file size 1.5GB
- Number of Posts(Question + Answers) = 29499662 ($\sim$ 30 million), file size 45GB
- Number of Votes = 98928934 ($\sim$ 99 million), file size 9GB

Total dataset size of $\sim$ 55GB

To process this huge dataset, I broke each file into multiple smaller files and processed one at a time. As the whole big file was not fitting into the memory of system.

8 Visualization and Results

8.1 Simulation Results

Below results validates the correctness of the implementation of model.

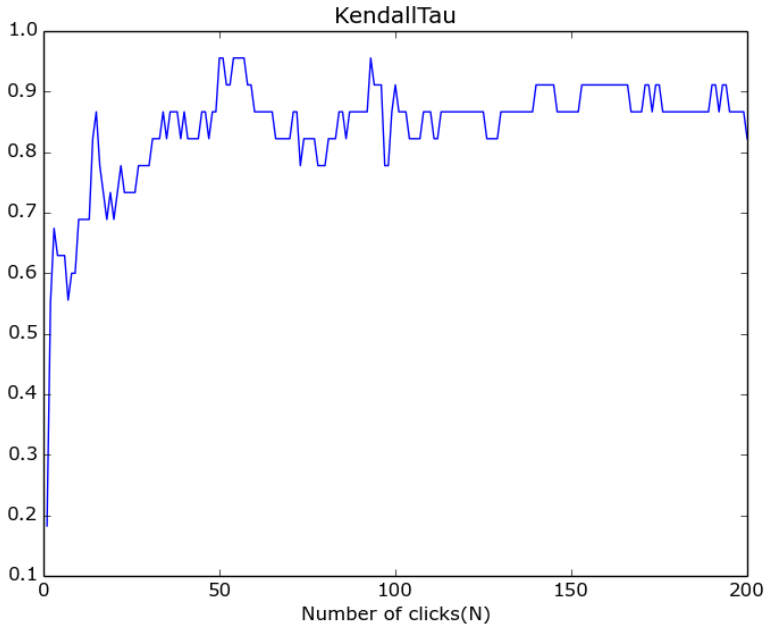

Figure 8: Sample Graph

8.2 Real data TimeLine View

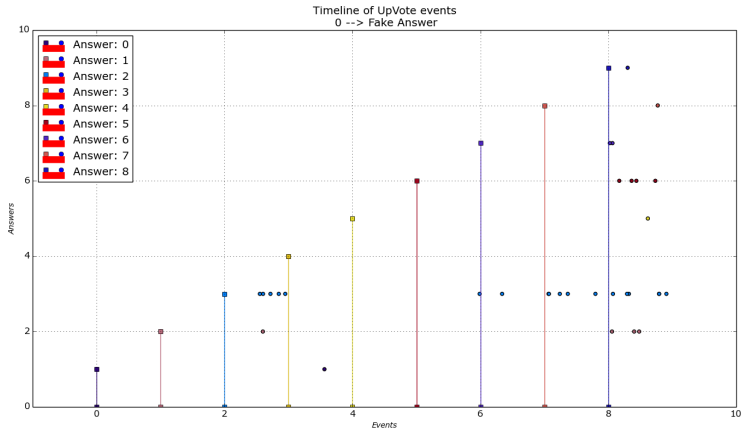

Figure 9: Sample Graph

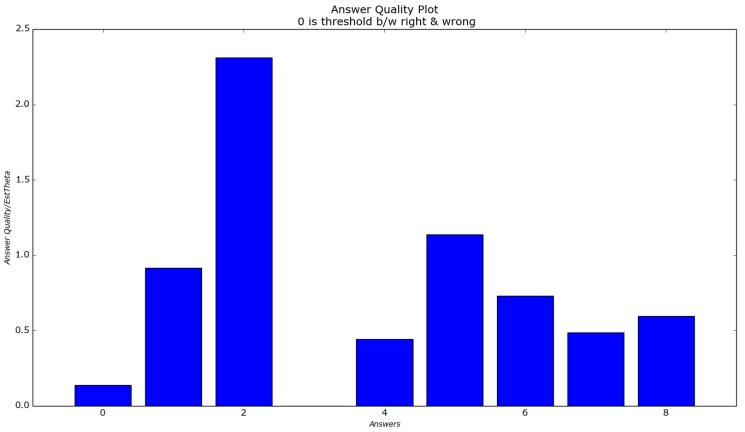

Figure 10: Sample Graph

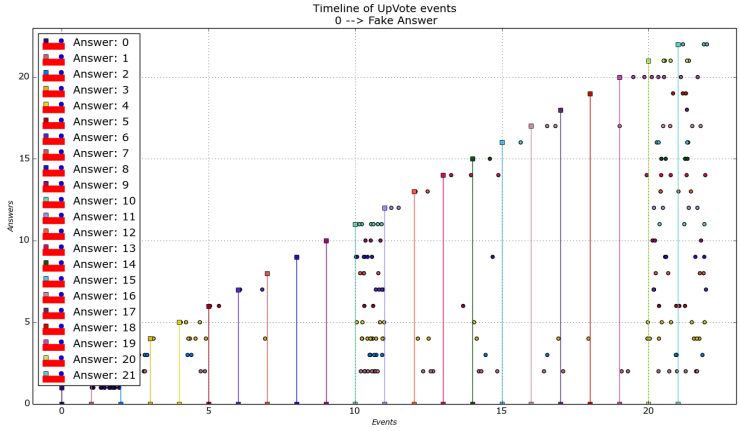

Figure 11: Sample Graph

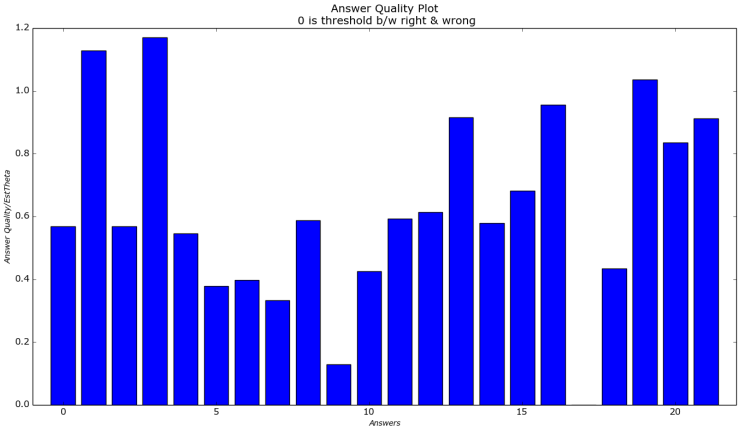

Figure 12: Sample Graph

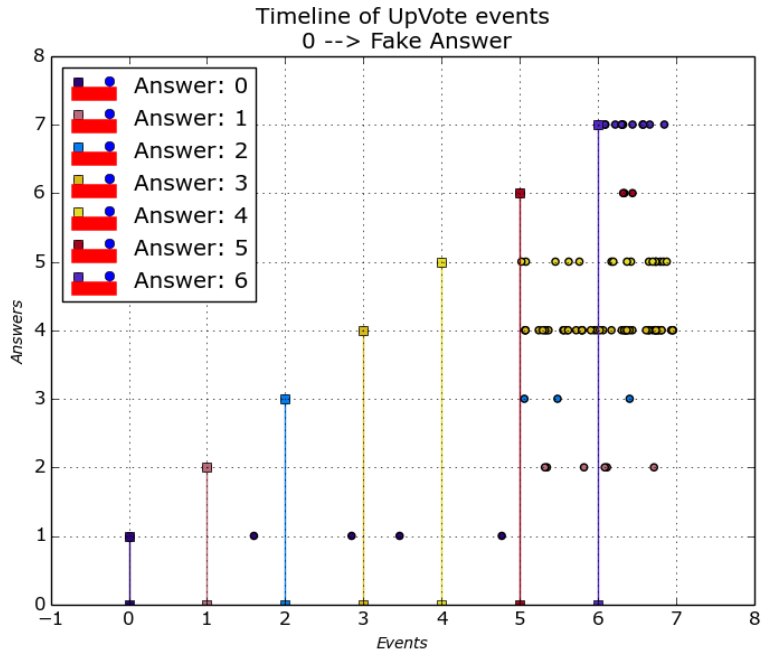

Figure 13: Sample Graph

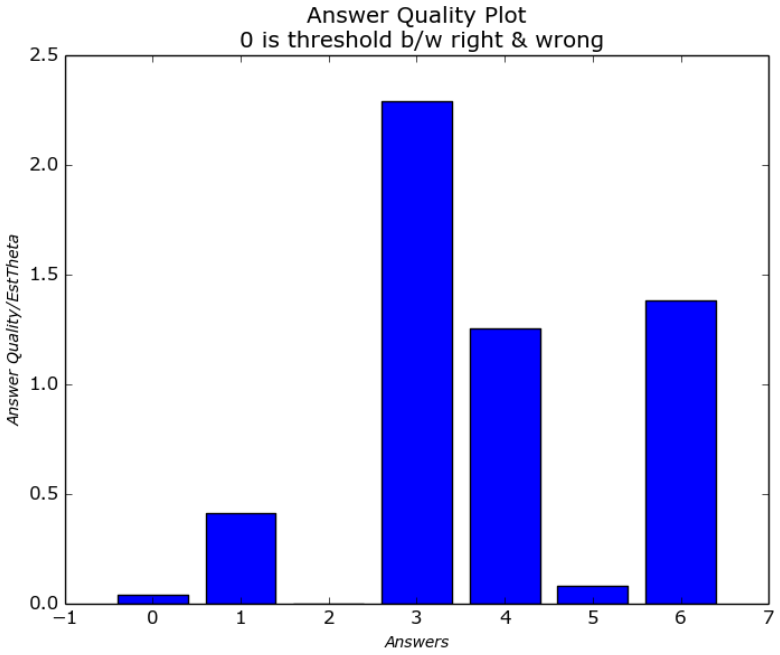

Figure 14: Sample Graph

8.3 Statistics from BIG DATA

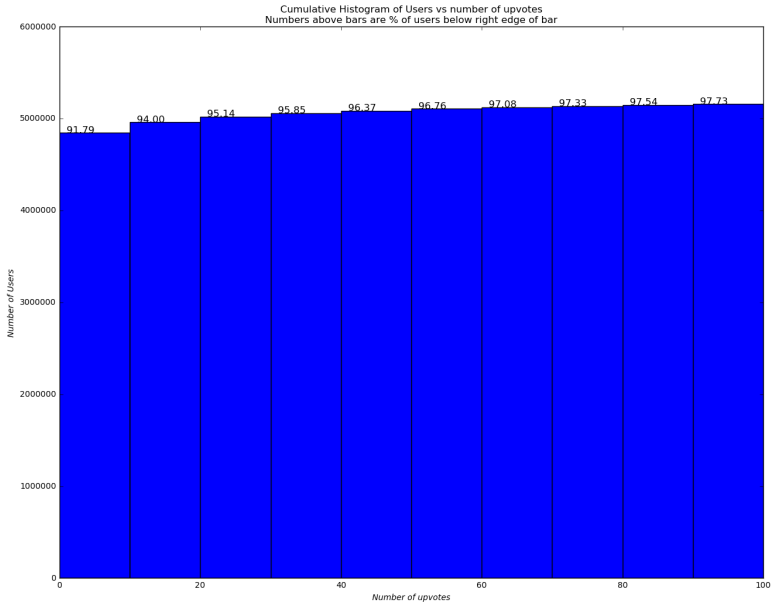

Figure 15: Sample Graph

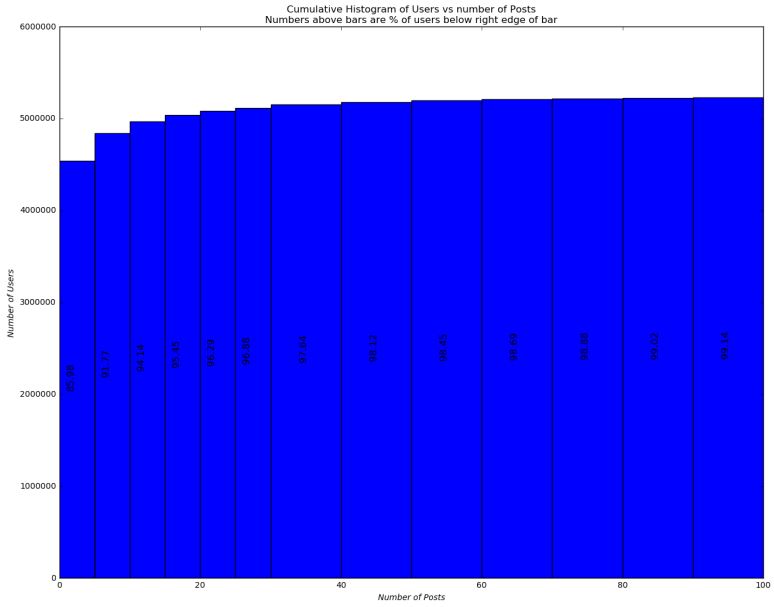

Figure 16: Sample Graph

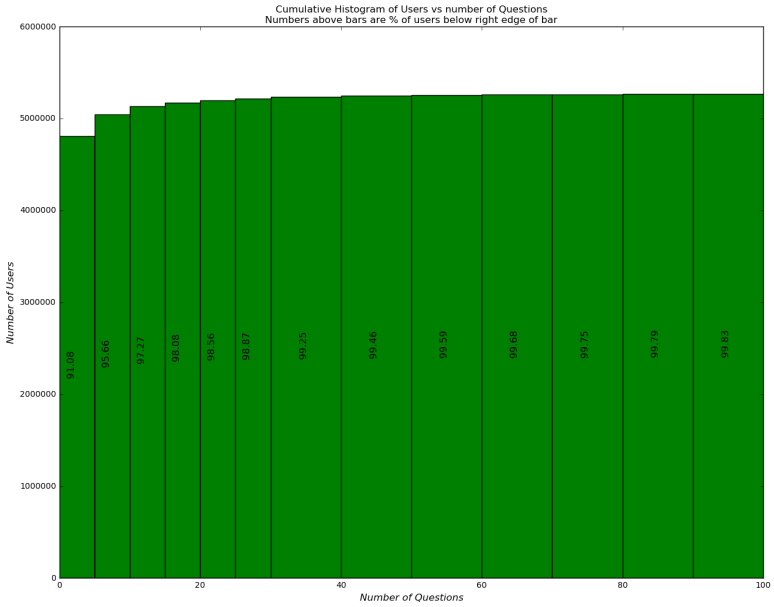

Figure 17: Sample Graph

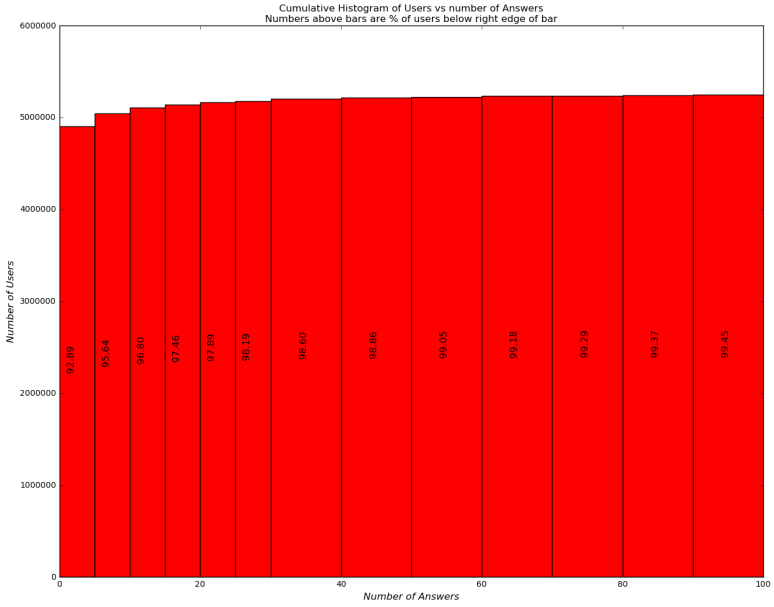

Figure 18: Sample Graph