StackOverflow Data Analytics

Ashwin Bhide, Chaitanya Bapat, Nidhi Menon, Sneha Venkat, Vaibhav Tendulkar

Problem Definition

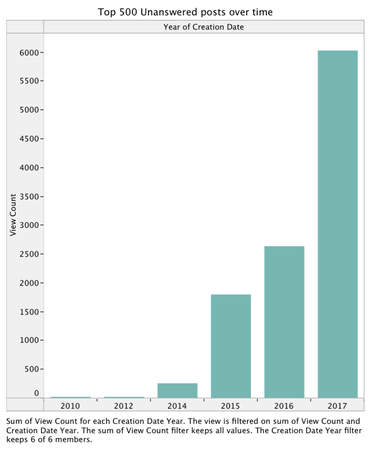
StackOverflow is an online question answering community for programmers providing rapid access to knowledge and expert peers

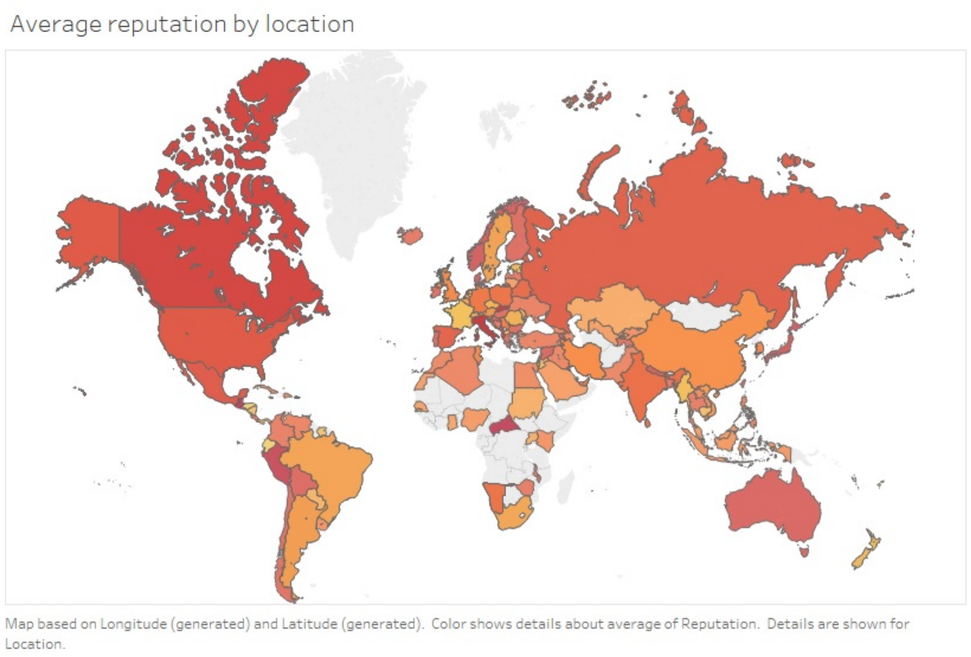
1. To analyze StackOverflow to find trends in data, & understand **community user engagement** dynamics.
2. To design a **question routing** & **expert-recommendation** system

Motivation

There is an increasing trend in number of unanswered questions on StackOverflow.

Currently, StackOverflow displays **unanswered questions by tags** instead of routing questions to specific users.





Intuition

Our approach is to identify experts in particular domains and route relevant questions to them. To understand user engagement, we used graphical visualization techniques. Routing questions to experts will help the questioner get the correct answer quickly.

Evaluation

Algorithm suggests 10 users which are most likely to answer the question in test set satisfactorily. Compare the results with users who actually answered questions on StackOverflow. Tag-based match and higher reputation scores indicate a better question routing algorithm.

Data

1. StackOverflow [2mn+ records]: UCI ML repository
2. StackExchange API Data Explorer using Python Requests.

Feature Engineering

1. Based on question post, computed features like *Readability Score, Number of External Links, Word count, Number of Code Fragments,* etc
2. Constructed answerer vector based on A*nswerer id, Reputation, Upvote count, Downvote count, Number of Accepted Answers.*

Innovation

1. Inclusion of **Readability Score**
2. **Expert recommendation** system

Approach

1. K-means clustering

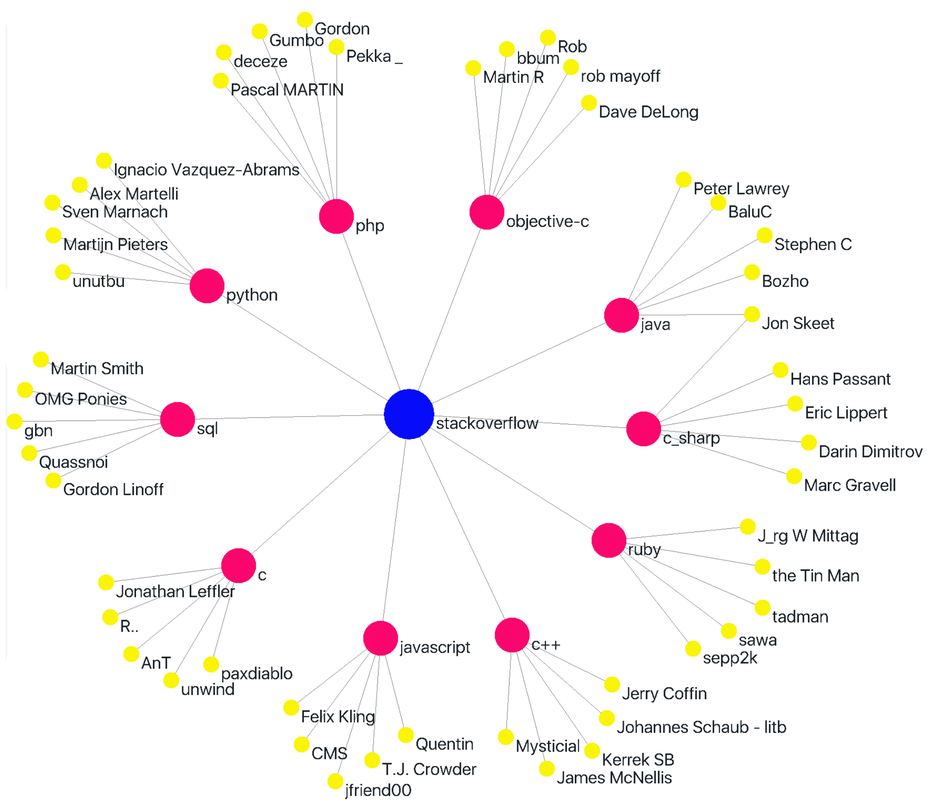
Clustering questions based on **tags** and identify **top N** users who have answered most number of questions in the cluster.

2. Random Forest classifier

Construct feature vector based on question and answerer vectors and feed it to algorithm. Use **cosine similarity** as a measure to find fixed set of questions most similar to the new question vector. Highest probability question-answerer pair is chosen.

Learning Phase - Question vector + User vector

Evaluation Phase - argmax(Σ feature_score)



Results

Upon querying an unanswered question (e.g. PHP) our algorithm was able to predict an expert user (PHP domain-specific) who also featured in Top 2% with higher reputation score. Unanswered questions like these should be targeted to such experts.