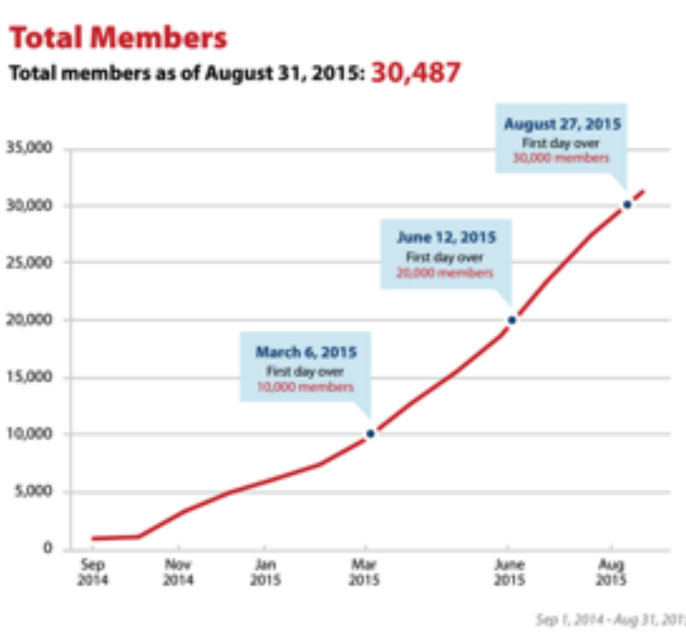
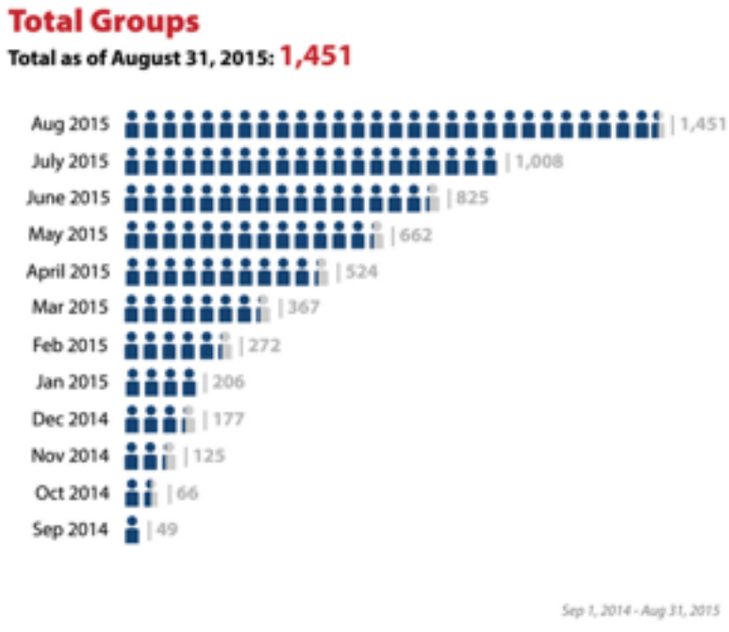
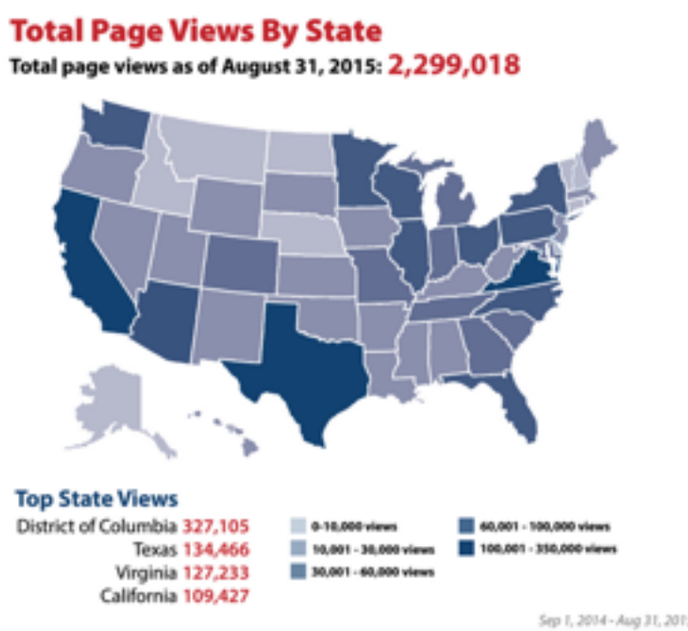
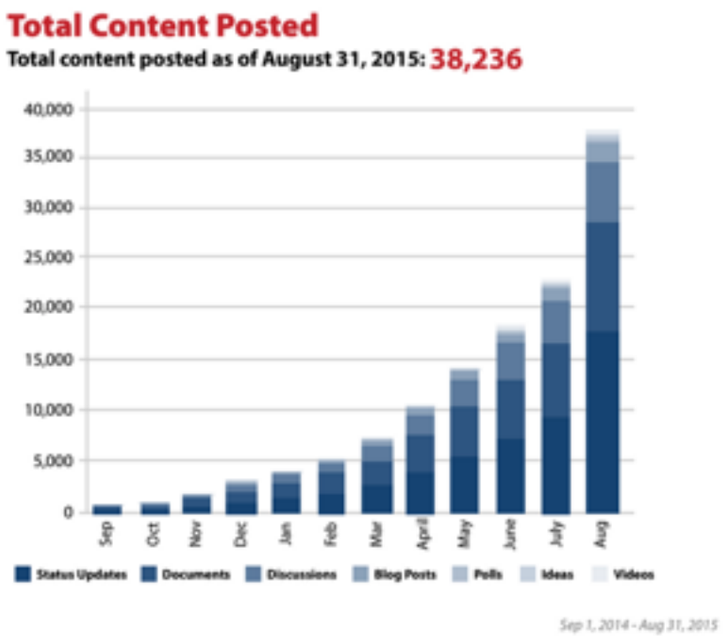
The Problem:
True Adoption of Enterprise Collaboration Software

- Enterprise Collaboration Software is only valuable if people actually use it.
- Achieving true adoption of Collaboration Software is a lot harder than you might guess.
- Significant investment to get adoption: software investment, training, internal marketing, etc.
- Gamification presents an opportunity but thus far is based on intuition rather than data









The Question(s)

- Can we predict which users will “adopt” based on behavior in the first month?
- If so, can we use the model to inform our process for on-boarding new users? Interpretability of model would be an excellent bonus.
- Clarifying Decision 1: Use deployment that is just over 1-year old with ~12,000 users who joined platform between 6 and 12 months ago. Deployment is ongoing and focus is shifting form acquiring users to retaining users.
- Clarifying Decision 2: What is definition of “adopt”? Lots of data analysis
    1. Consumer: at least 5 consuming actions / month (more than 1/week)
    2. Contributor: at least 5 contributing actions / month
    3. Changed definitions to not be mutually exlusive

The Data

- Underlying SQL database captures a ton of data about users and the activity they perform in the system (think google analytics level data without blinding).
- User profile data is voluntary and thus data completeness / quality is limited; ongoing side investigation for better profile data (e.g. ip address giving location).
- Other activity in Excel logs (training participation, demo participation); this data proved to be too time consuming to scrub so this got dropped. Waiting for an intern support.

Data Cleaning / Gathering

Normalizing data to show the following per user (big effort):

User Profile

- UserId
- Title (free text self-description)
- District (general region, 5 values after cleaning)
- Location (fixed list after some cleaning but more than 200 unique values)
- Occupation (fixed list after some cleaning with about 50 unique values)
- Phone provided (bool)
- Biography provided (bool)

User Activity

- ~ 2mm rows
- User Id
- Activity Type (~40 types)
- Date/Time Stamp
- Additional metadata (dependent on type)

SQL Magic;
Python too slow

User Activity

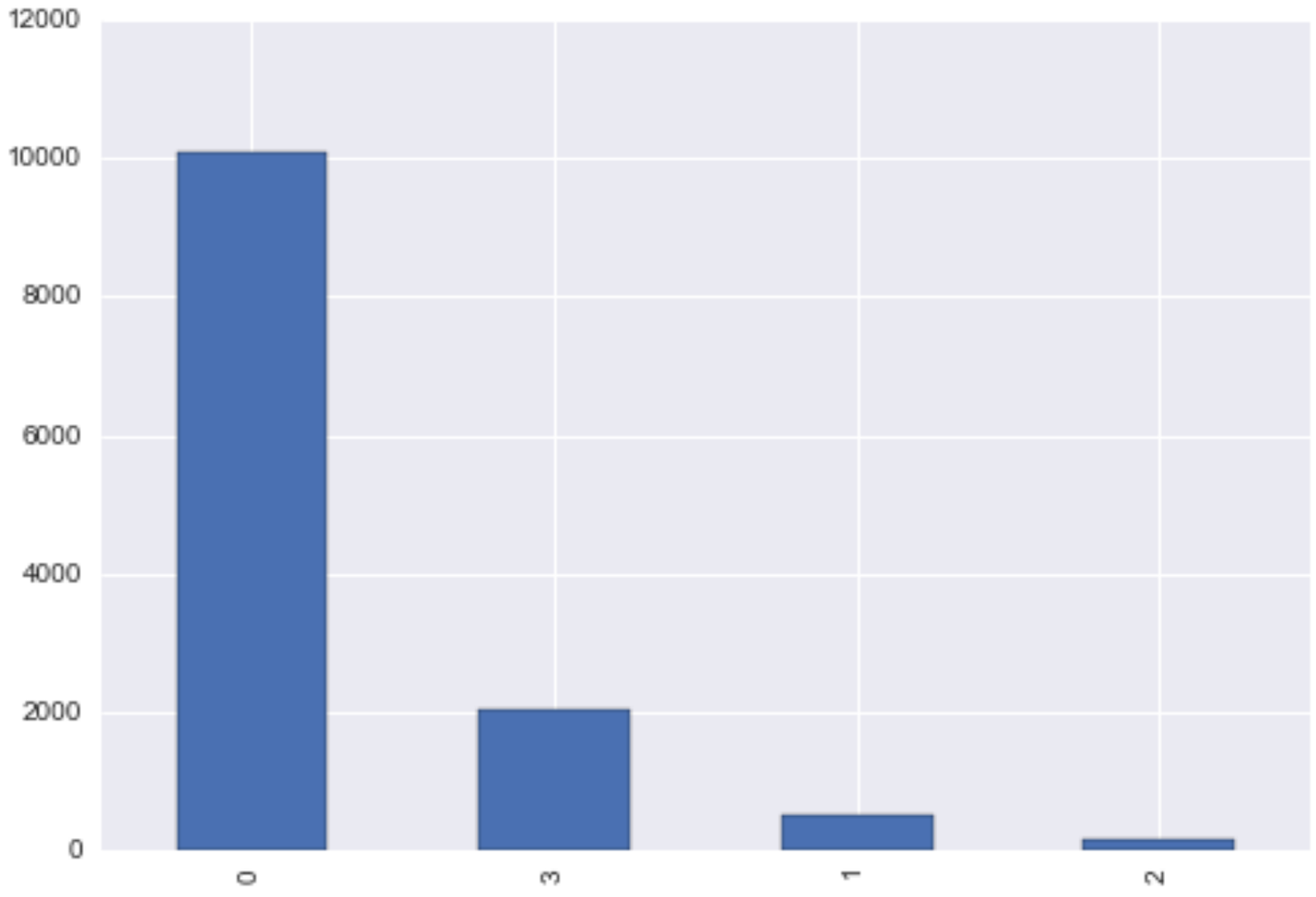
- User Id
- Activity Type (~40 types)
- Month 1 Count
- Month 6 Count

Data Transformation & Feature Engineering

- SQL:
  - initial filtering of data, (this proved harder than anticipating)
  - anonymizing,
  - standardizing categories,
  - bucketing activity counts into months (processor intensive)
- Pandas:
  - Pivotting to create columns for each activity type
  - Join and merge to combine multiple datasets
  - Factorized boolean features
  - One-hot encoded categorical features (this explodes the feature count)
  - Categorized 6-month activity counts and summed
  - Generated binary response variables for consumer and contributor

High-level Analysis and Null Model

- Consumer: 80% accuracy predicting No
- Contributor: 83% accuracy predicting No



Modeling: Started w/ Random Forest - Consumer

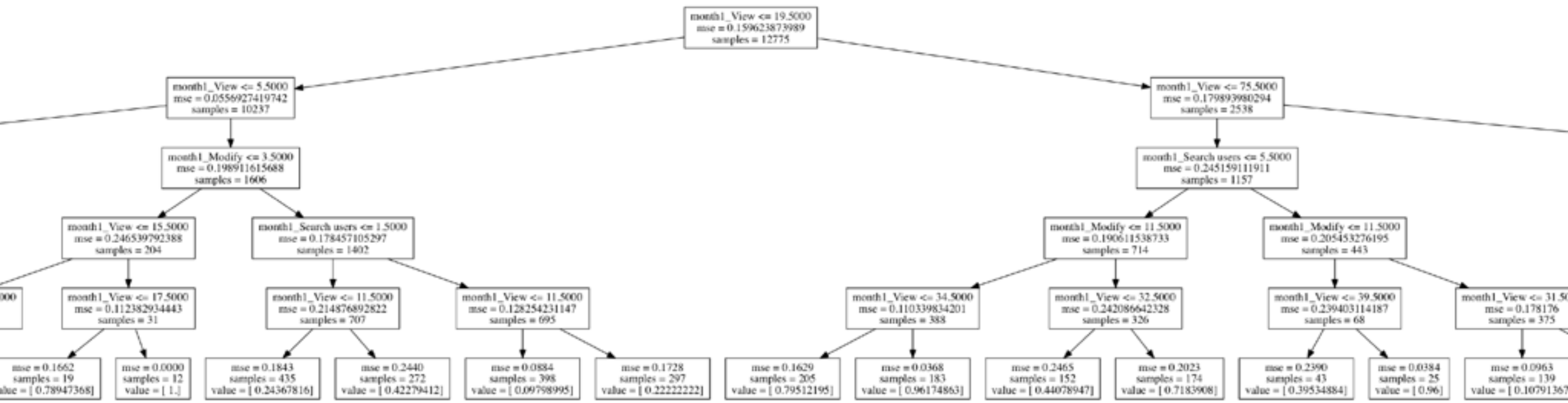
- Tuned model parameters to get to a 92.9% accuracy (killer slow); .60 OoB score
- Used Train-Test-Split to get AUC .962 (is this valid?); very sensitive; not so specific.
- Most interesting items: looking at feature importances; those with importance over mean were; limiting to just important features didn't really improve things:
    - Views
    - Modifies
    - Logins
    - Follows
    - User Searches
    - Creates
    - Downloads
    - Content Searches
    - Spotlight Searches
    - Associates
    - Approves
    - Likes
    - Providing Phone

Modeling: Started w/ Random Forest - Contributor

- Tuned model parameters to get to a 92.1% accuracy; .60 OoB score
- Used Train-Test-Split to get AUC .947; ok sensitivity (72%) great specificity (96%)
- Most interesting items: looking at feature importances; those with importance over mean were (basically same as consumer; some reodering):
    - Views
    - Modifies
    - Creates
    - Logins
    - Follows
    - User Searches
    - Downloads
    - Content Searches
    - Spotlight Searches
    - Approves
    - Likes
    - Associates
- Limiting to these features didn’t improve metrics in any significant way though

Modeling Continued: Used "Best" Features from RF to do Logistic Regression and Decision Tree

- Performed logistic regression using best features; results were forgettable.
- Generated Decision Tree using best features:



Modeling Fun: Text Analysis on “Title” Multinomial Naive Bayes

- I was missing out on Text Processing
- Quickest thing to look at was “Title” which people use to describe what they do; dropped all observations without this and analyzed.
- Created Document Term Matrix; not actually that many unique terms (for y records)
- Ran through models; very little improvement over null model (. 1 % and AUC of)
- Attempted to ensemble w/ Random Forest model but couldn’t figure out how to model for records with no “Title” (significant %).

Immediate Next Steps

- A new month of data is about to be available: opportunity to predict with truly out of sample data and further refine model
- Add in features for social data, membership in key groups, associations with key users (after investigating this; flaws in pulling data out of SQL were discovered).
- Update model to predicting whether a user is going to drop-out based on current rolling months worth of activity; enables team to re-target users before they drop
- Rework overall system to bucket data by week as opposed to by month.

Far Future Questions

- Predicting whether an employee is going to quit based on activity (or changes in activity) could enable focused interventions from managers.
- Predicting employee quality (e.g. review score) based on activity could allow for interventions from manager or coach; alternatively show employees some form of their own predictive score and “model employees”.
- Predict whether an employee matches criteria for another position; use data about position changes to predict whether an employee would be eligible for an open position to mine data more effectively for internal recruiting.