Predicting H1-B Visa Application Outcomes
Final Project, STAT471 Fall 2014
(Casey) Juanxi Li - Individual

Goal
The goal is to determine whether the approval or denial of an H1-B visa application is a random lottery, or whether there are identifiable factors that can be used to predict the decision.
The response variable to be predicted is "Status", with possible values CERTIFIED or DENIED.

Other values such as WITHDRAWN or HOLD exist, but are thrown out of the dataset for the sake of simplicity.

Data and Summary
The data was obtained from Enigma.io. Access to their data is free but requires account signup. H1-B data from 2006 – 2014 can be obtained by searching for "visa applications". All fields from the H1-B application, minus identifiable information about the employee, appear to be represented as columns. Each dataset is roughly 35-40 features x 350k – 400k rows.

For the sake of workability and processing power, I have limited my project to a subset of 2014.

Methods
There is a large imbalance between the response values of CERTIFIED vs DENIED in the original dataset: roughly 97% vs 3%. Since the goal is classification, the dataset was randomly undersampled with replacement to an even split (or else a classifier which simply denies all applications would have 97% accuracy). This was also necessary to prevent my computer from crashing when attempting to handle 400k rows of data.

Continuous or numerical variables to which I applied logistic regression are **application date (month and year)**, **annual salary, time until decision**, **duration of employment requested**, and **full vs part-time**.

The **employer name, role description, and role title** were concatenated into a "bag of words", on which I ran a Naïve Bayes text classifier (packages *tm and e1071*).

Lastly, zip code data was mapped onto the continental US using the *zipcode* package in R and plotted using *ggmaps*.

Findings
Though histograms suggest some general trends relating to time and salary, logistic classifiers generally had 30 – 40% error rates, indicating that non-text information on the H1-B application cannot reliably be used to predict an outcome.

The Naïve Bayes text classifier consistently yielded a ~36% testing error, indicating that the corpus of text between accepted and rejected applications are not entirely identical and that key words may be identified through repeated testing.

Finally, zip code mapping visually indicates a slightly less dense concentration of denials in urban areas, so the next step may be to locate a dataset which classifies zip codes as urban vs non-urban, and add that as a feature to the classifier.

Overall, however, no findings are robust enough to indicate that the H1-B application process is obviously rigged or discriminating against any type of application in particular.

Packages Used

ggplot2, pROC, MASS, tm, wordcloud, e1071, ggmap, zipcode

Preliminary Notes

On Using Excel and Reproducing my Output

The gargantuan nature of the dataset meant I had to use Excel for certain cleanup and formatting tasks. All .csv files from every point in the process are included in the folder; the R code should run without interruptions as long as the packages above are installed.

Imbalanced Data

Conventional knowledge is that the USCIS issues a cap on H1-B visa applications every October (65,000 for 2014). Several factors explain the 97/3 imbalance between CERTIFIED and DENIED applications in each year's dataset of approximately 350,000 applications, including the following (sourced from the USCIS website or emails with Enigma.io staff):

- US universities, government, non-profit research institutions, and affiliated organizations are not subject to the H1-B cap
- There is an additional 20,000 cap for students pursuing Masters degrees and certain advanced programs
- An employee who was counted towards an H1-B cap in a previous can apply for extension and/or a change of employment terms without being counted into the cap again for six years

Errors

The data includes obvious intentional errors (i.e. bogus salaries with random digits or meaningless strings of random characters in the company name), which warrant further investigation.

The data also contained unintentional errors such as typos/missing spaces, or the salary reporting unit being mismatched with the pay amount (i.e. a salary of $70,000 being reported as hourly), or missing decimal places, indicating that the data was likely entered using Optical Character Recognition or (sloppily) by hand.

Errors were cleaned by hand when a solution seemed obvious (i.e. changing a salary frequency from hourly to yearly if it seemed to make sense based on the role description), or the line was deleted entirely if no sensible explanation was obvious.

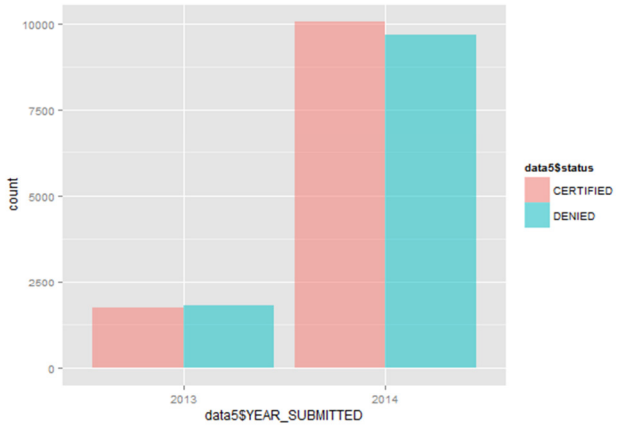
Methods

Due to the nature of the dataset (extremely large and in variable formatting), cleanup and the creation of variables such as "days until decision" were created manually in Excel. Ideally, undersampling, cleanup, and analysis would have been repeated several times, but the manual nature of the task prevented this from being a possibility.

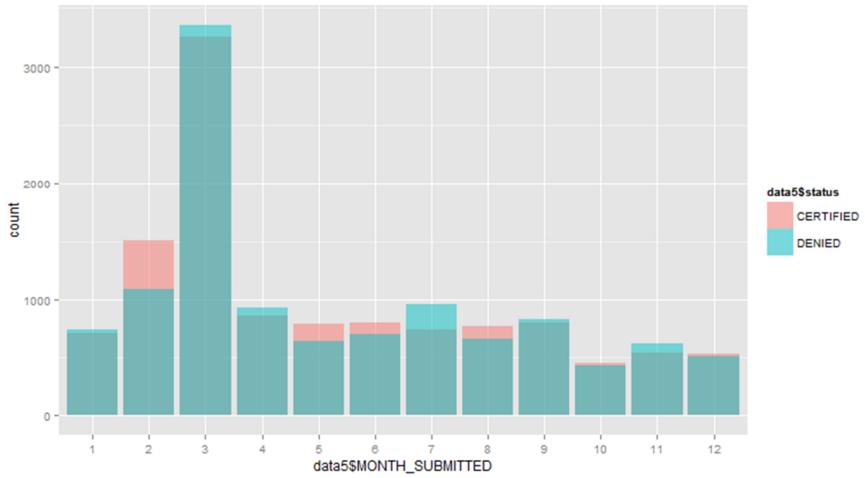
Charts and Summary

The original dataset contains about 400k rows, split 97/3 between Certifications and Denials. Certifications were randomly undersampled to obtain a dataset of about 23k rows, split roughly 50/50. All charts reflect this undersampled data set.

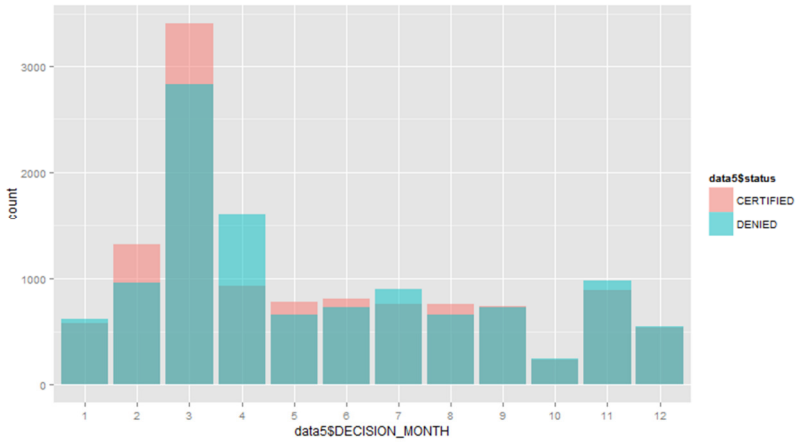
2013 vs 2014 submissions (for the FY 2014 cap):



By Month Submitted:

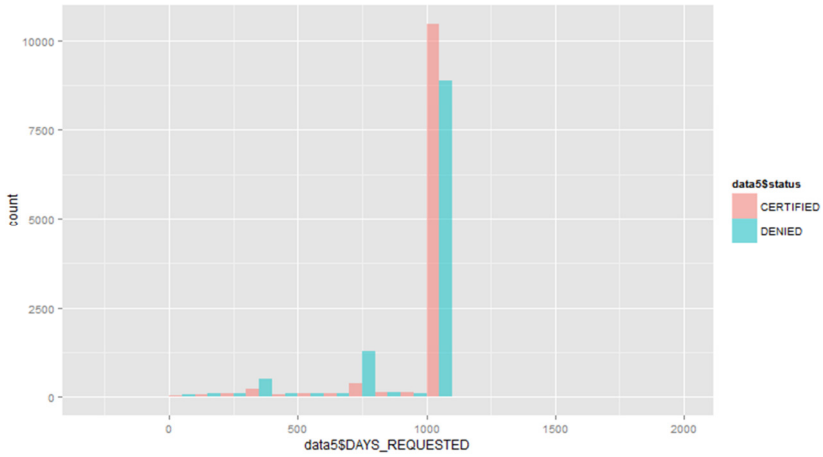


By Month Decided:



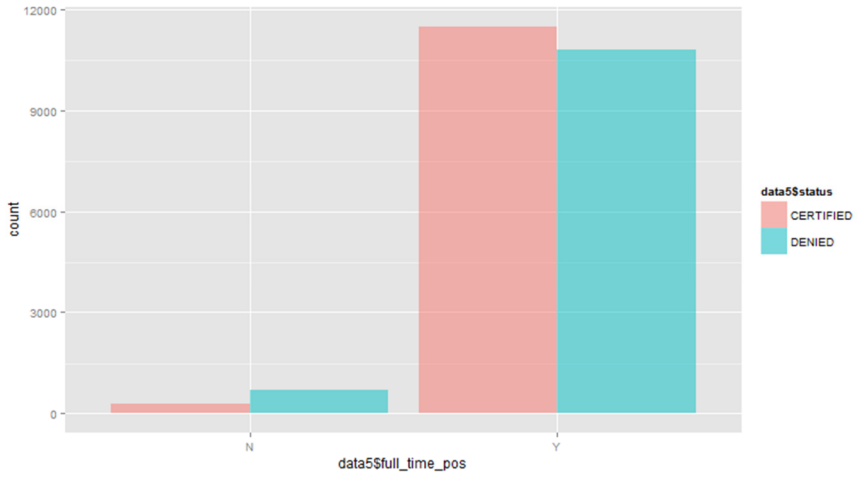
***Note** that the spike in March submissions and decisions may be due to the fact that the earliest start date for an application which counts towards the 2015 cap is the beginning of FY2015, or October 1st 2014. The application deadline for this start date is **April 1st, 2014**.

Days Requested



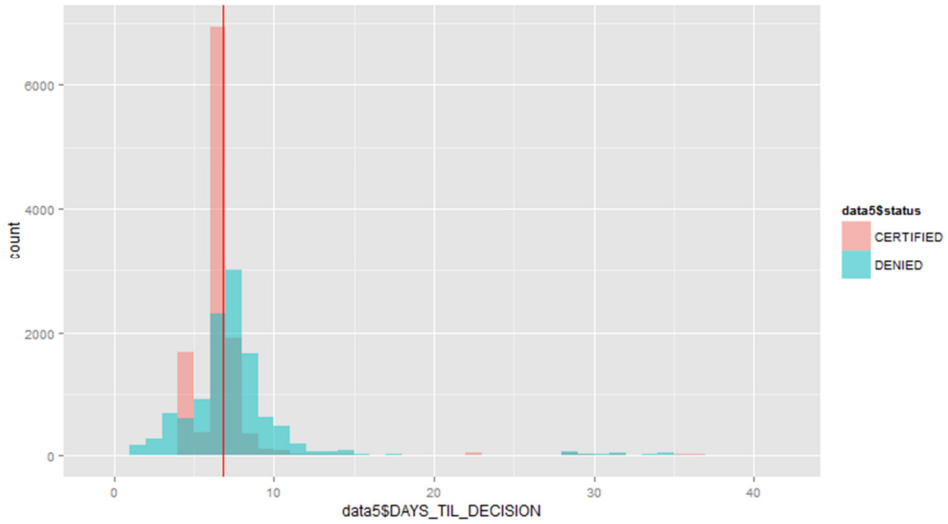
Days requested tend to come in multiples of 365 days, up to a maximum of 3 years.

Full-vs Part-Time:



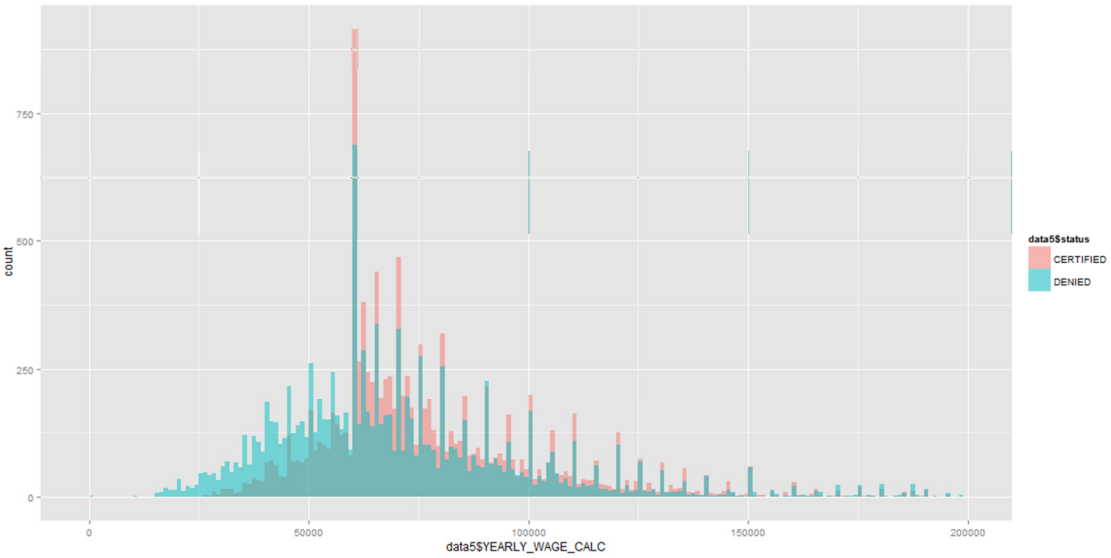
There appear to be a higher portion of certifications for full-time applications.

Days Until Decision:



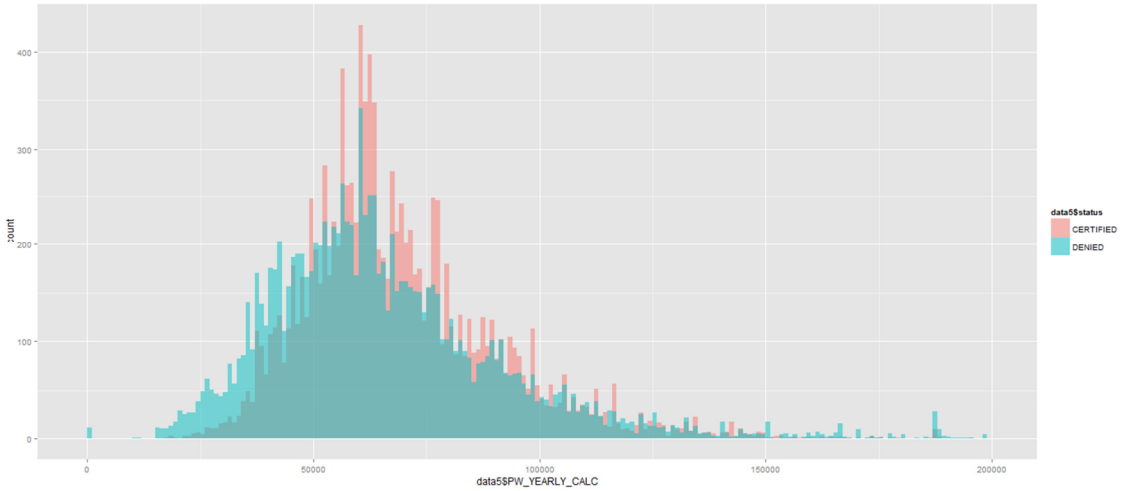
The mean time is 6.7 days from application submission until decision. Approved applications appear to cluster more tightly around the 6.7 day mark, with a small spike at 4 days.

Annual Wage for the Employee being sponsored:



Overall, salaries for approved applications appear to be higher and right-skewed.

Annual Prevailing Wage for the Position



A similar pattern.

Classification: Logistic Regression

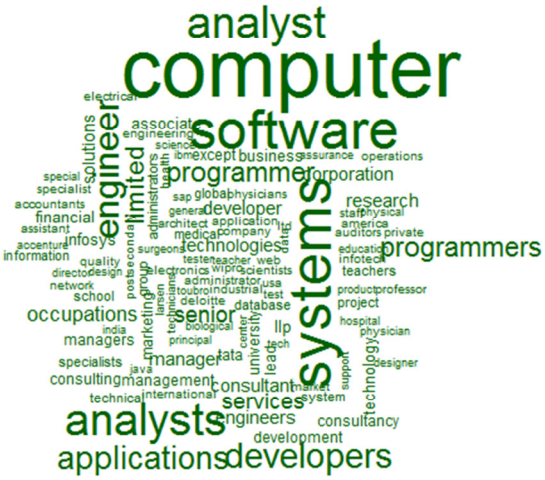
Despite the interesting patterns from the charts above, logistic regression yielded a 33% error rate, with the classifier giving a 23% false negative rate (denying an application when it was accepted) and a 44% false positive rate (accepting an application when it had in fact been denied):

```
> cm=table(data5$status, fit1.pred) # confusion matrix
> cm
           fit1.pred
            CERTIFIED DENIED
  CERTIFIED      9066   2714
  DENIED         5081   6414
> cm=table(data5$status, fit1.pred) # confusion matrix
> cm
           fit1.pred
            CERTIFIED DENIED
  CERTIFIED      9066   2714
  DENIED         5081   6414
>
> training_error = (cm[1,2] + cm[2,1])/nrow(data5)
> training_error # 33%, not very good
[1] 0.3349087
>
> table(data5$status)

CERTIFIED    DENIED
    11780     11495
```

Next Step: "Bag of Words"

The dataset was stripped down to the response (Certified or Denied), with the company name, role description, and role title mashed together into a "bag of words". A random sample had to be taken within the already undersampled subset due to computational limits (at about 10,000 rows on my laptop, which I bought 4 years ago).

The words were stripped down, and simple text clouds were created from the corpus of text in each category to see if there were any obvious patterns (there weren't):



Naïve Bayes Text Classifier

A Naïve Bayes text classifier was built and tested on samples of the data ranging from 2000 – 10,000 rows (anything above would cause my computer to hang), a testing/training split of 0.25 – 0.50, and with infrequent words being knocked out from anywhere between words appearing once to words appearing less than five times. Laplace smoothing for words in the testing set that did not appear in the training set made no significant difference.

Testing error was stable around 36%, with type I errors (certifying an application when it was in fact denied) being more common.

```
> cm_NB = table(test_set$status, test_pred)
> cm_NB
           test_pred
            CERTIFIED DENIED
  CERTIFIED      1306    624
  DENIED          719   1101
>
> testing_error = (cm_NB[1,2] + cm_NB[2,1])/((cm_NB[2,1] + cm_NB[2,2])+(cm_NB[1,2] + cm_NB[1,1]))
> testing_error
[1] 0.3581333
>
> # testing error hovers around 36%
>
> type_1 = cm_NB[2,1]
> type_1 # false +ve: predicted accepted, actually denied
[1] 719
> fp_rate = type_1/(cm_NB[2,1] + cm_NB[2,2])
> fp_rate # ~40%
[1] 0.3950549
>
> type_2 = cm_NB[1,2]
> type_2 # false -ve: predicted denied, actually accepted
[1] 624
> fn_rate = type_2/(cm_NB[1,2] + cm_NB[1,1])
> fn_rate #~32%
[1] 0.3233161
```

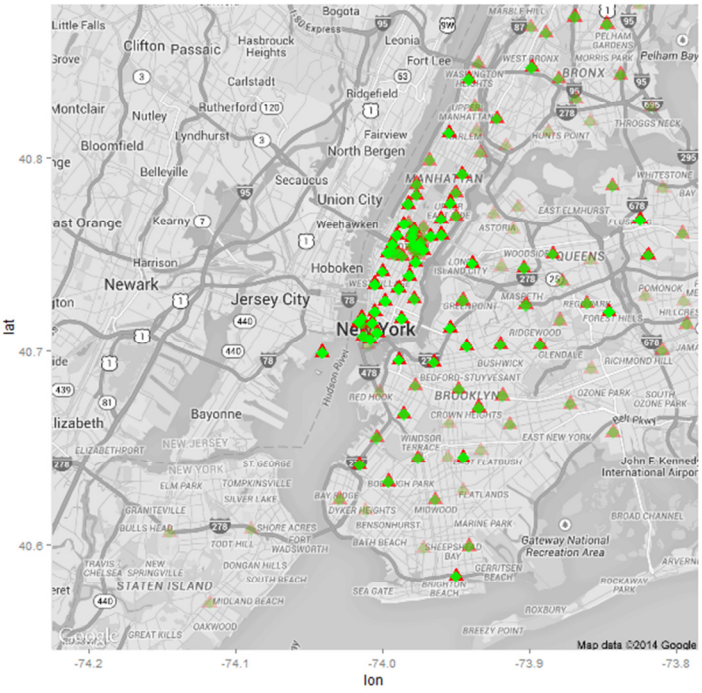
Here are some sample conditional probabilities from the classifier. Repeating this classification many times while keeping an average of conditional probabilities may be a way to reveal certain influential keywords; however, I have yet to figure out how to code a loop to do so.

```
                        university
factor(train_set$status)         No        Yes
               CERTIFIED 0.97040498 0.02959502
               DENIED    0.93421053 0.06578947

$vice
                        vice
factor(train_set$status)           No         Yes
               CERTIFIED 0.996884735 0.003115265
               DENIED    0.991776316 0.008223684

$video
                        video
factor(train_set$status)           No         Yes
               CERTIFIED 1.000000000 0.000000000
               DENIED    0.993421053 0.006578947

$web
                        web
factor(train_set$status)           No         Yes
               CERTIFIED 0.995327103 0.004672897
               DENIED    0.998355263 0.001644737
```

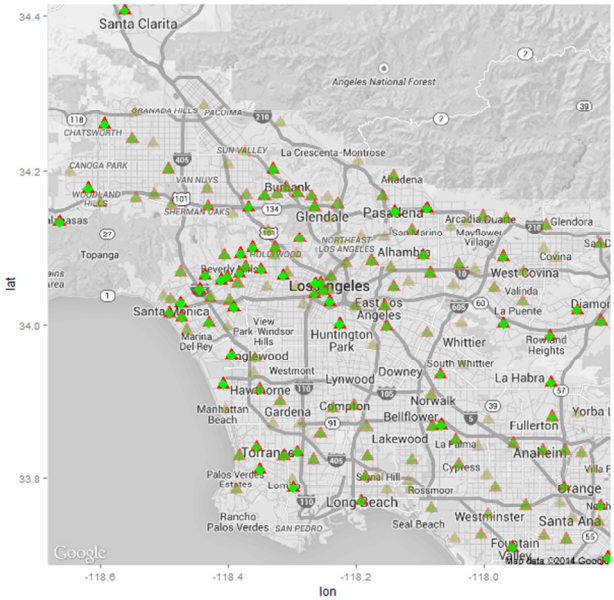
Zip Code Data

Lastly, I wanted to see if anything interesting could be pulled out of the zip code data in the project. Green indicates certification, red is denial. Points were set to 90% transparency, so a bright area indicates a buildup of applications.

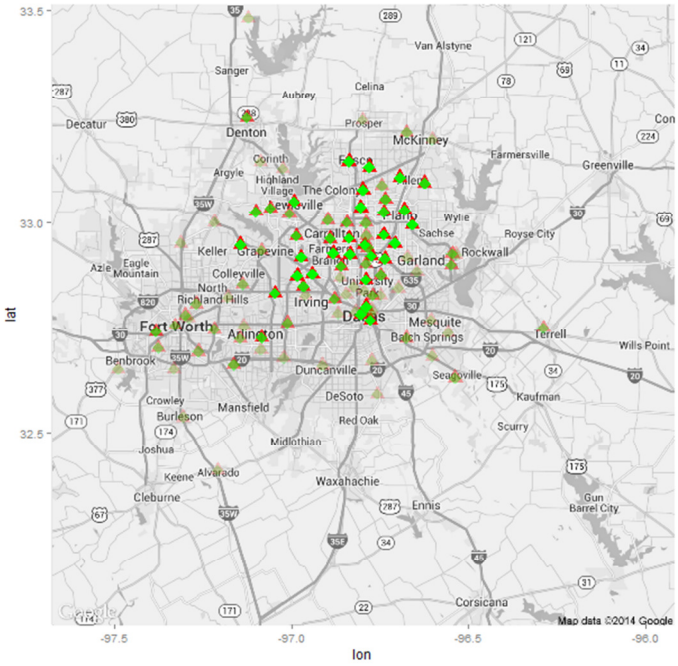
A map of applications around metro New York:



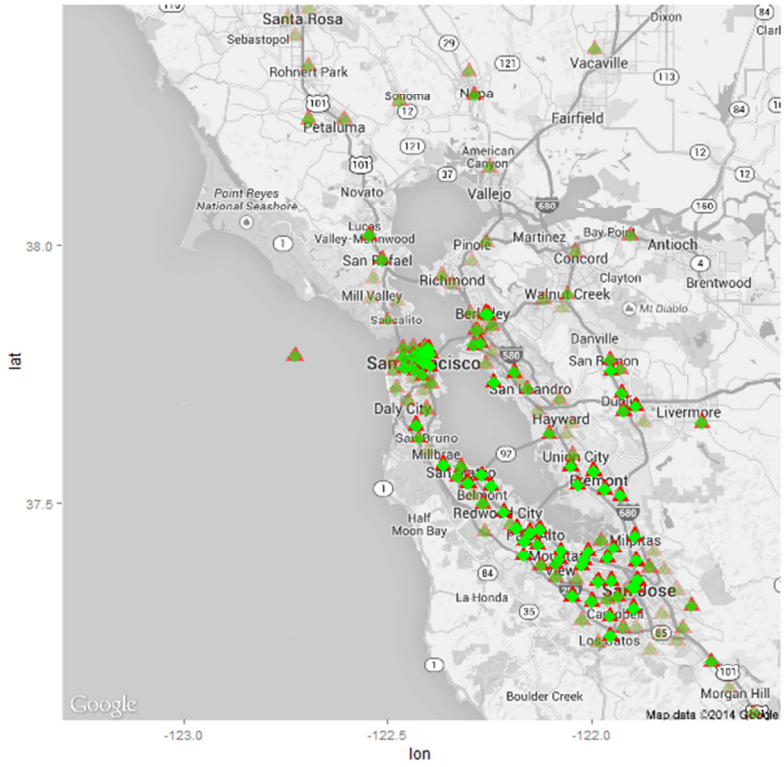
Los Angeles:



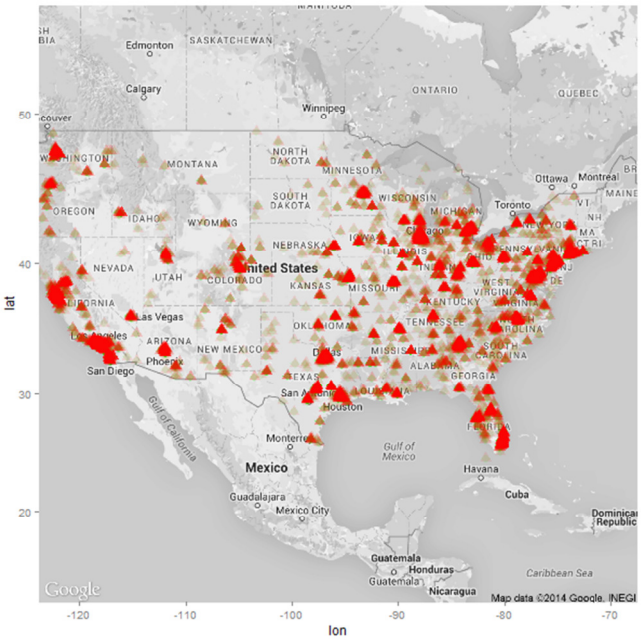
Dallas:



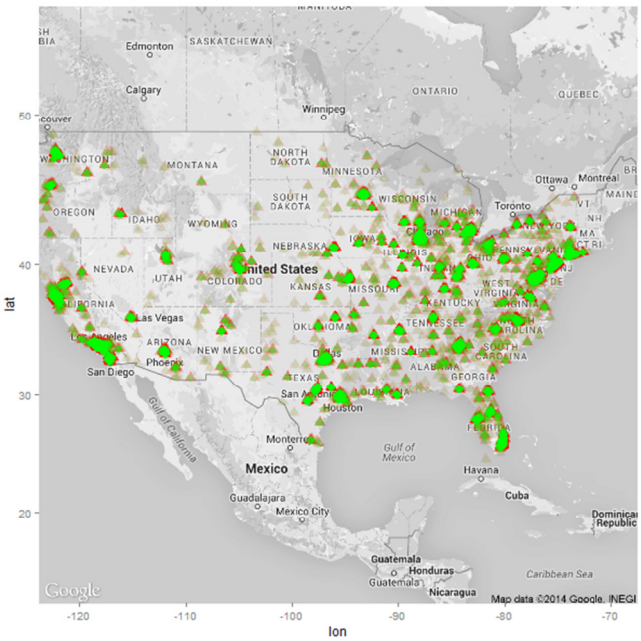
San Francisco:



Overall United States, with denials layered on top:



Certifications layered on top:



It may be an illusion of color, but denials appear to be less tightly clustered in major urban areas than acceptances. Next step would be to somehow bucket the application zip codes by population density and/or per capita income (as measures of proximity to a major urban area) to determine if there are any interesting relationships with application acceptance.

Conclusion and Further Explorations

Eyeballing the charts seems to indicate that some features are related to application outcome, such as:

- Higher salary
- Proximity to April at application time
- Full-time position
- Short # of days until decision made

However, the logistic prediction model's 33% error rate indicates that the current methods should be repeated with multiple undersampling from the rest of the data to ensure that this wasn't due to chance. Unfortunately, this is complicated by errors in the data and hardware limits.

The **Naïve Bayes text classifier** consistently achieves a 36% testing error, indicating that there may be some meaningful information buried in the application text. As this implementation of Naïve Bayes is extremely simple (i.e. word count is disregarded and all words are assumed to be independent of one another, even though "business analyst" means something very different than "systems analyst"), refinements on Naïve Bayes and other methods of text classification (SVM, Maximum Entropy) are worth exploring.

Zip code visualization indicates there may be a relationship between applications and urban centers/population density worth investigating as well.

The bright side

There's no glaring evidence to indicate that the H1-B lottery is systematically discriminating between applications in any obvious way. Any patterns need to be tested repeatedly against the entire dataset to ensure they did not arise from chance.

Thanks for reading! Any suggestions for improvement or next steps would be greatly appreciated. ☺