KIVA Expiration

Team 22
Kaung Mon Khin & Dylan Connor
5/3/2018

Abstract - Background and Impetus

- **Organization:** Kiva crowdsourced microfunding
- **Problem:** Identify loans that are not going to be funded
- **Method:** Binary classification ML task
- **Motivation:** People on Kiva are mostly from underdeveloped countries/regions. Getting funded for their projects is potentially life-changing. Kiva and its partners can use this information to either focus on either
  - Candidates who are more likely to be funded
  - More effectively market loans that have higher likelihood of expiration

kiva

Data Sources and Cleaning

- Kaggle
  - Individual Loan data (~650k) / Partner ID
- Kiva API
  - Additional individual loan data – Contains status
- University of Oxford – MPI data

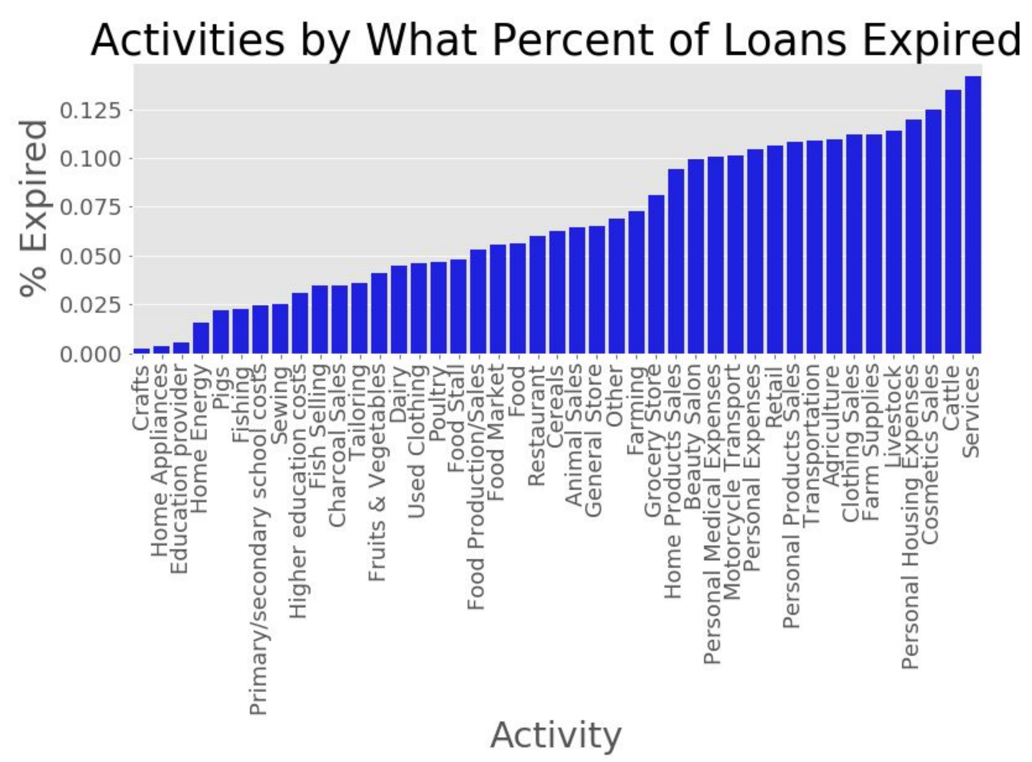
kiva

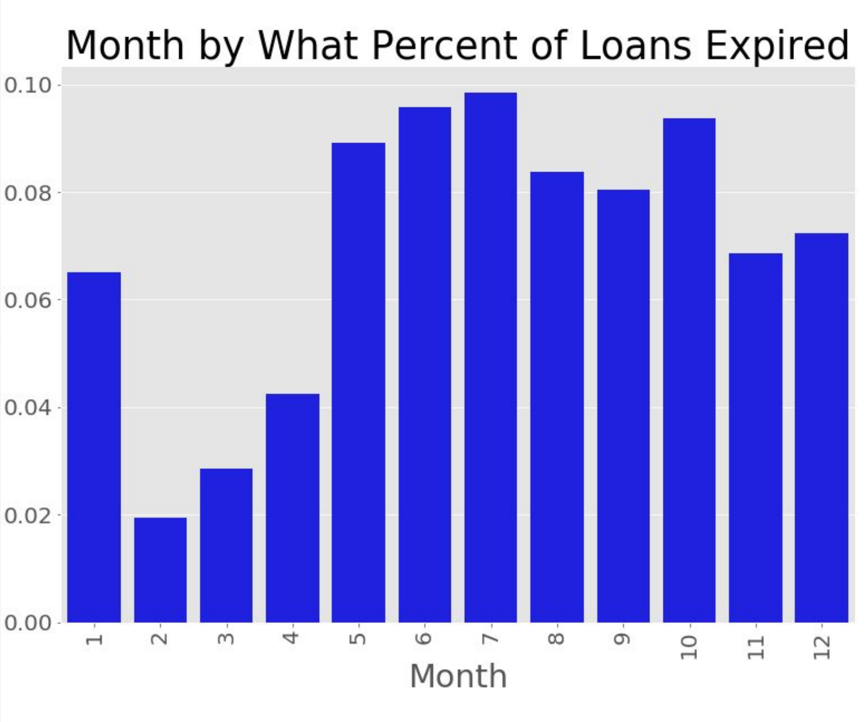
Feature Selection

Feature	Description
Status	If loan expired
Loan Amt	$ amount of loan
Activity	Subcategory for loan
Sector	Segment of business
Country	Country of origin
PartnerID	Unique ID for partner
Term	Length of loan
Repay Int	Regular or Irregular

Feature	Description
Language	Native Language
MPI	Poverty Index
Year	Year of loan
Month	Month of loan
Gender Counts	Count of males and females
Days to Expiration	Expiration - start days

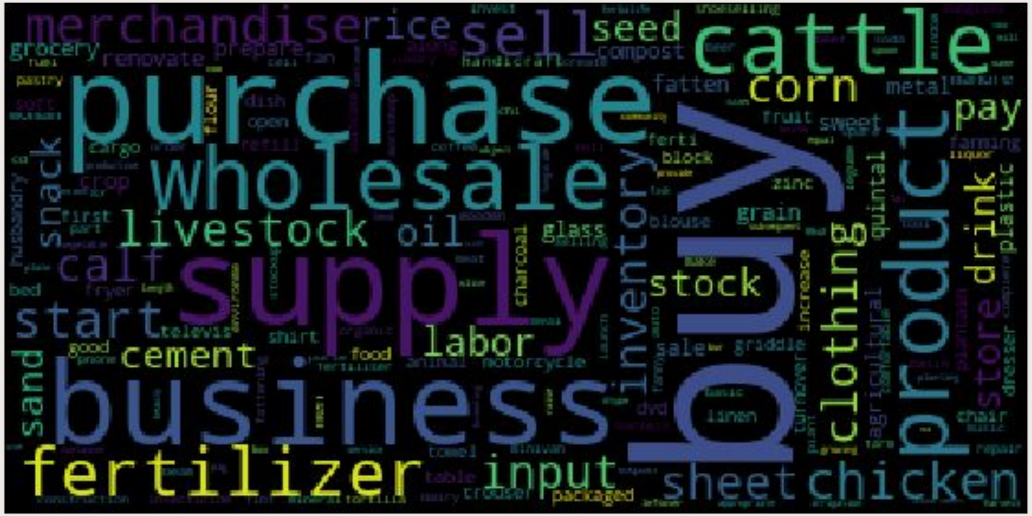
Exploratory Data Analysis – Features





kiva

EDA – Word Cloud



Expired Status Loans

Funded Status Loans

Modeling Preparation & Initial Modeling

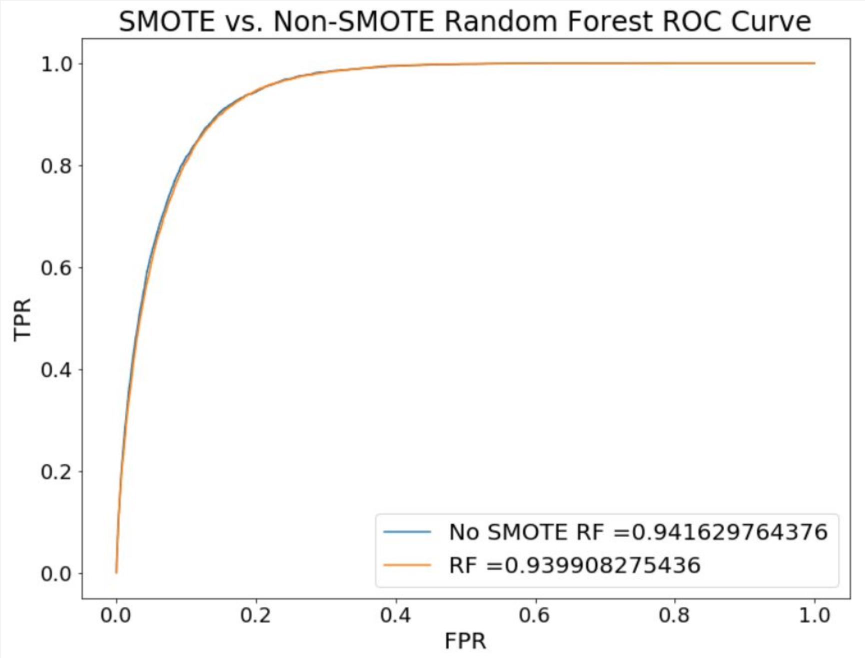
- One hot encoded categorical variables keeping top factors with >.5% or population, labeling else as “Other”
- Normalize with min-max methods
- Parse the text out of descriptions, lemmatize them, remove stopwords and, break into sentences and words

Modeling Preparation – Unbalanced Data

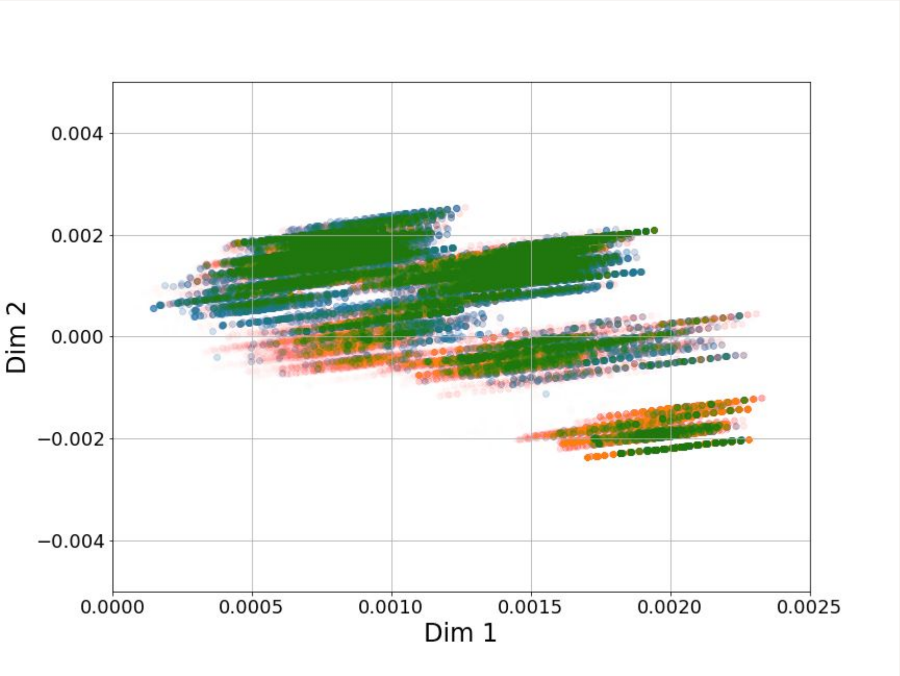
- Use SMOTE to deal with unbalanced data

Normal	0	1
0	108162	1291
1	5330	2054

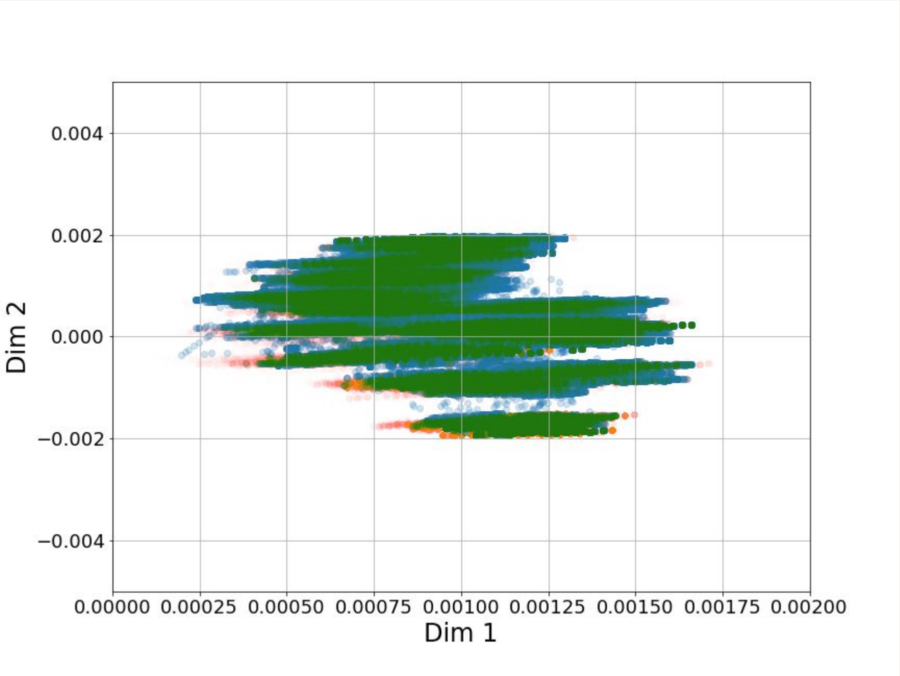
SMOTE	0	1
0	100123	9330
1	1748	5636



Modeling Preparation – PCA Comparison



Unbalanced Data

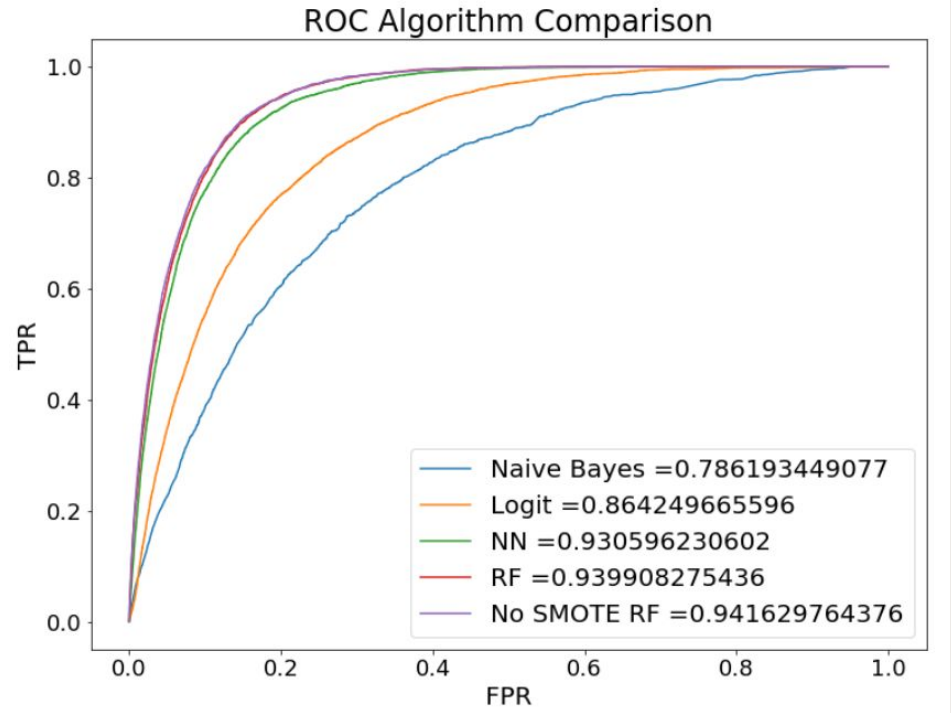


Balanced Data (SMOTE)

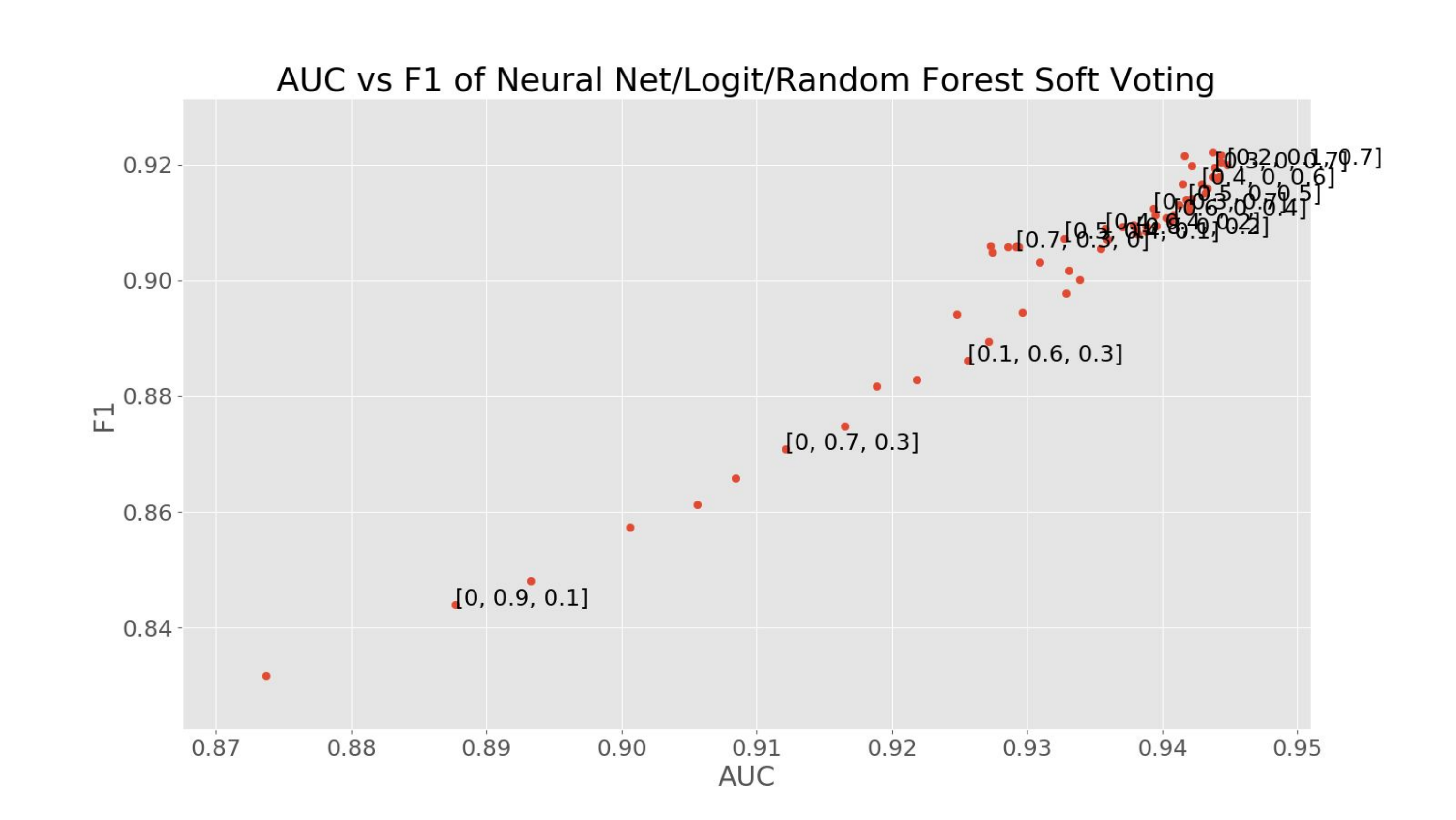
kiva

Results

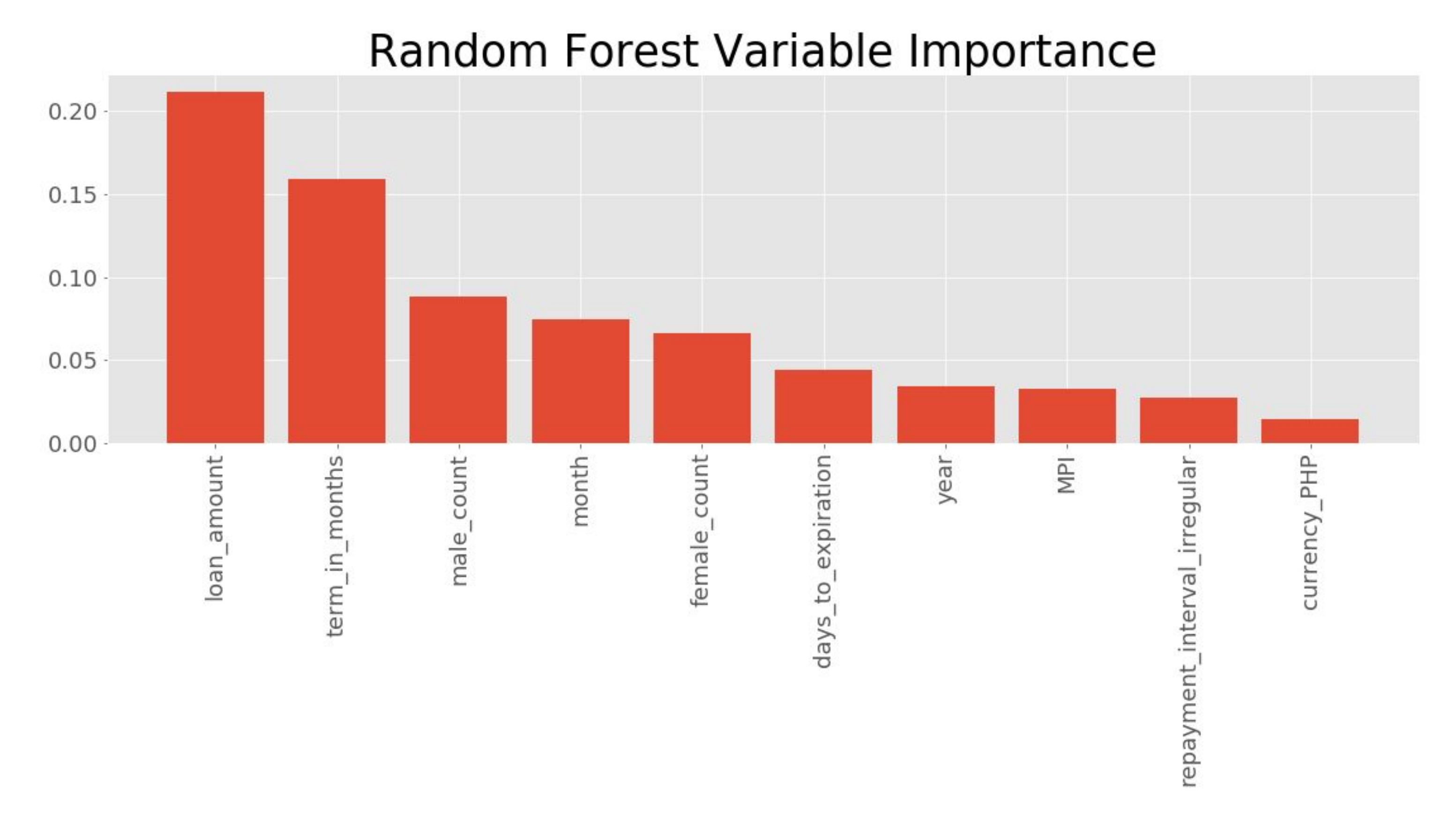
Model	F1	AUC
Naïve Bayes	.76	.78
KNN	.91	.78
Logistic	.83	.86
RF (no smote)	.93	.94
RF (w/ smote)	.92	.94
Neural Net	.93	.93
Voting Ensemble	.91*	.85*



Results - Soft Voting Probabilities



Results - Interpretability (SMOTE RF)



Results - Text Analysis

- We did our analysis without balancing the data for the text analysis
- Basically, very POOR results!
- Further research needed into generating a “SMOTE-like” data for text

Model	AUC
Simple RNN	.52
LSTM	.5196

kiva

Conclusions

- SMOTE is awesome
- The importance of metric selection
- Specific algorithms tendency to overestimate either minority or majority class
  - SVM and Logit vastly overestimate minority class

Difficulties and Next Steps

- Memory and computational issues (SVMs with kernels, cross-validation on RF and NN)
- Ensemble voting cross-validation
- Generative text modeling
- Better feature level interpretations and correlation analyses
- Map geographies better and get more granular geocode sub-region details