Analysis of kiva.com Microlending Service

Hoda Eydgahi Julia Ma Andy Bardagjy

December 9, 2010

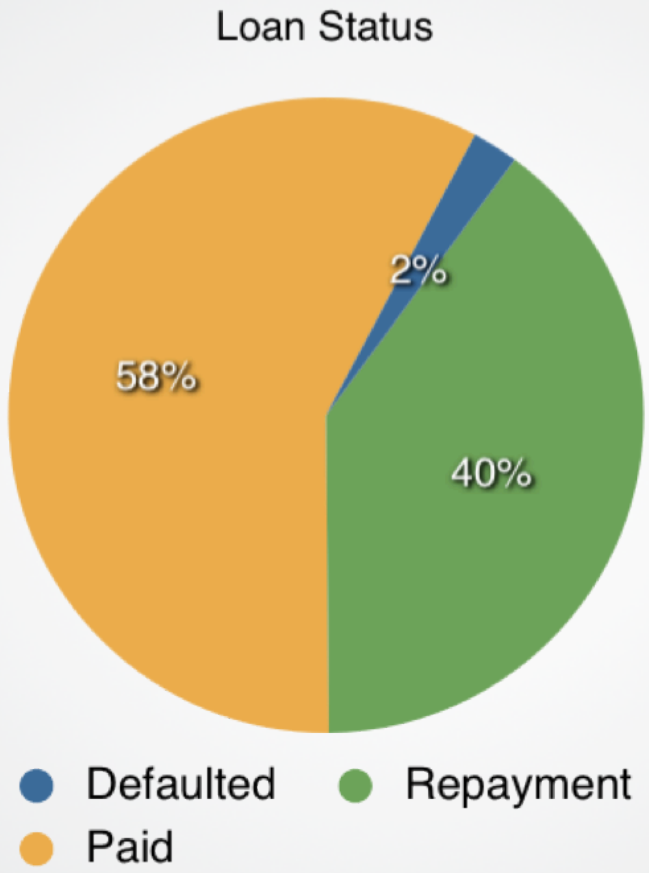
MAS.622j

What is Kiva?

- An organization that allows people to lend small amounts of money via the Internet to micro-finance organizations in the developing world.
- Local micro-finance organizations help local business people to post profiles and business plans on Kiva.
- Once online, lenders select profiles and business plans to fund.

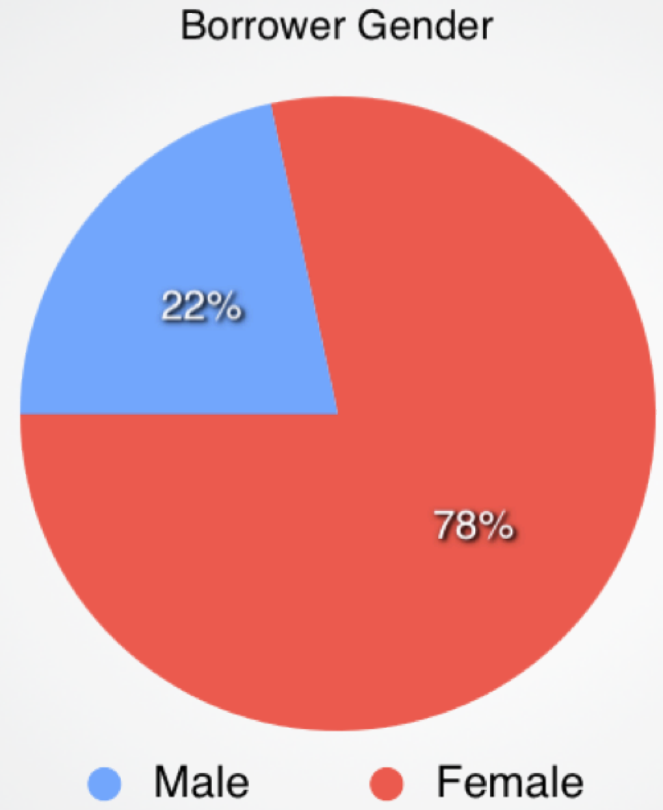
Data Profile

kiva.com microlending service

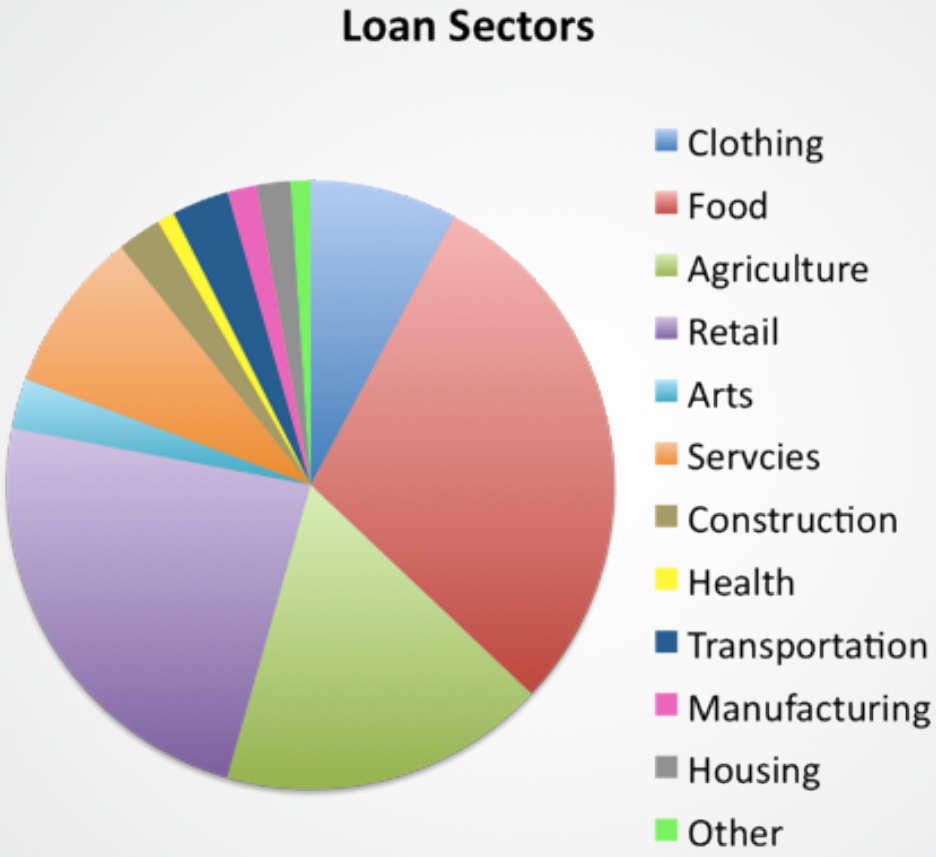


LOANS

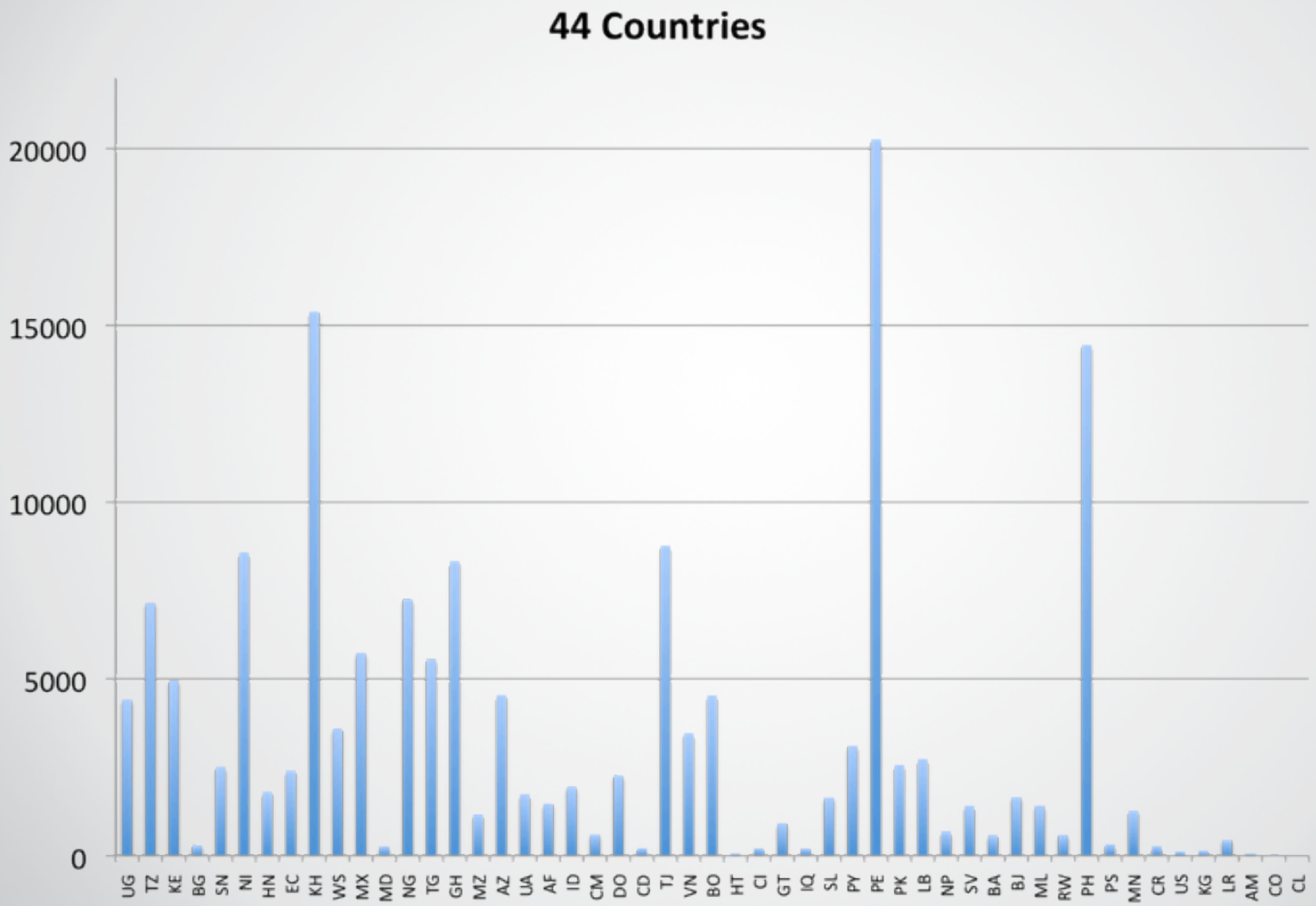
LOANS



LOANS


Loan Sectors
Clothing
Food
Agriculture
Retail
Arts
Servcies
Construction
Health
Transportation
Manufacturing
Housing
Other

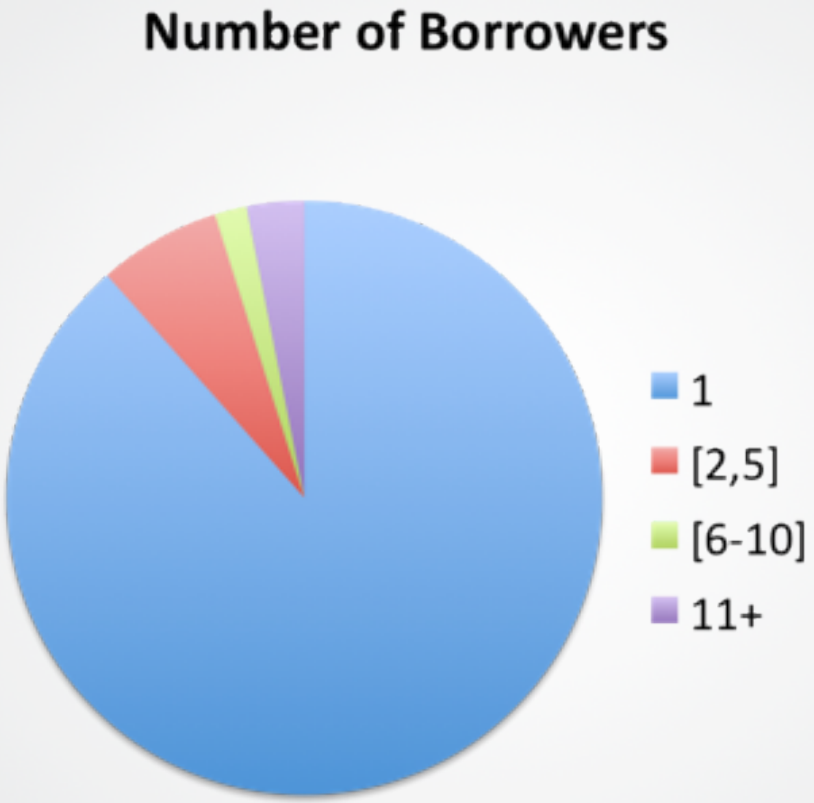
LOANS


44 Countries
20000
15000
10000
5000
0
UG TZ KE BG SN NI HN EC KH WS MX MD NG TG GH MZ AZ UA AF ID CM DO CD TJ VN BO HT CI GT IQ SL PY PE PK LB NP SV BA BJ ML RW PH PS MN CR US KG LR AM CO CL

LOANS


Debbie
Alex Jane Anthony
and
Carol Robin
Carolyn
George
Megan
Rachel Donna Beth
Bob
Aaron
William
Scott Emily Kelly Janet
Pat Andrea
Shannon
Larry
Sandra
Michelle Judy
Marie Anna
Family Jessica Erin Gary Julia
Jack
Angela Joseph
Steve Jill
David
Nick Sara
Stephen Ashley
Dave
Jason
Richard
Kathryn Ron Kate Ken
Joshua
Robert Rob
Mark
Lynn
Patricia
Susan Katie
Sarah
Ellen Christopher
Chris
Nancy
Julie Maria
Ben
Jon
Laura
Sue
Andy
Mary
James
Margaret
Rick
Linda
Don
Ian Heather Josh
Christine
Suzanne
Jonathan
Lauren
Elizabeth
Michael
Daniel Kathy Cynthia
Adam
Andrew
Bruce
Steven
Helen
Brian
Sharon
Jeffrey
Patrick
Martin
Doug
Kimberly
Jennifer
Amy
Sean
Jean Christina
Catherine
Jeremy
Paul Karen Jim Jan
Lori
Ann
Nicole Rebecca
John
Kevin
Wendy
Frank
Mike Peter
Barbara
Amanda
Cindy
Alan Pamela
Jeff Stephanie Matthew
Anne
Eric
Deborah
Leslie
Cheryl
Bill Melissa
Joe
Matt Ryan
Thomas Lisa
Kathleen
Charles
Joan
Greg Dan
Katherine
Diane
Justin

LENDERS


Number of Borrowers
1
[2,5]
[6-10]
11+

Questions

- Can we predict whether a loan will be paid back?
- Which feature(s) best predict the above question?

Classes

- Defaulted (prior probability 2%)
- In repayment, delinquent (prior probability 3%)
- In repayment, not delinquent (prior probability 37%)
- Paid (prior probability 58%)

Features

- Funded amount
- Paid amount
- Sector
- Number of borrowers
- Gender
- Length of use
- Length of description
- Number of periods
- Number of commas
- Number of languages
- Partner ID
- Number of journal entries
- Number of bulk journal entries
- Country code
- Rate of payback
- Percentage paid off
- Dictionary of words

Data Snapshot


```
{
   header: {
      total: "165452",
      page: 166,
      date: "2010-01-29T20:00:23Z",
      page_size: 1000
   },
   loans: [
      {
         id: 167286,
         name: "Francois",
         description: {
            languages: [
               "en"
            ],
            texts: {
               en: "Francois is a 31-year-old married man with 2 young children. He owns a jewelry shop where he makes and
               repairs jewelry. Today, Francois is asking to borrow $1,000 to buy tools and appliances needed to make a
               greater variety of products and thereby increase his sales."
            }
         },
         status: "in_repayment",
         funded_amount: 1000,
         paid_amount: 0,
         image: {
            id: 475617,
            template_id: 1
         },
         activity: "Jewelry",
         sector: "Retail",
         use: "to purchase tools and appliances for his business",
         location: {
            country_code: "LB",
            country: "Lebanon",
            geo: {
               level: "country",
               pairs: "33.833333 35.833333",
               type: "point"
            }
         },
         partner_id: 115,
```

Biggest Issues

- Data set is extremely large (165000+ points)
- Extracting features was computationally intensive
- Running algorithms was computationally intensive

Solutions

- Run full data set when possible
- Run code on computing cluster
- Reduce data set to a feasible size depending on the learning method
- Reduce data set size and adjust priors of classes (ie increase prior of Defaulted from 3% to 10%)

Methods

Prior Adjustment

- The dataset was too large to run most algorithms in a practical time
- Reduced the dataset from 165,000 to 30,000 data points
- Adjusted prior probabilities of classes to be more evenly distributed

	Defaulted	In Repayment, Delinquent	In Repayment, Not Delinquent	Paid
165,000 point data set	3%	3%	36%	58%
40,000 point data set	13%	13%	37%	37%

k Nearest Neighbor

- Algorithm that classifies by majority vote of k-number of nearest neighbors, determined by Euclidean distance
- Tested with odd k-values between 1-20
- Disadvantages
  - Classes with more frequent data points could dominate majority vote
  - Sensitive to features that are bad classifiers

k Nearest Neighbor

- Feature sequential forward selection was computationally intensive
- Use leave-one-out method of testing
- Reduce data set by an order of magnitude
  - 165000 → 10000 data points
  - Kept prior probability distribution the same

k Nearest Neighbor

- Best features for classification using SFS:
  - Funded amount
  - Number of commas
  - Length of description
- Classification accuracy: 49.2%

LDA

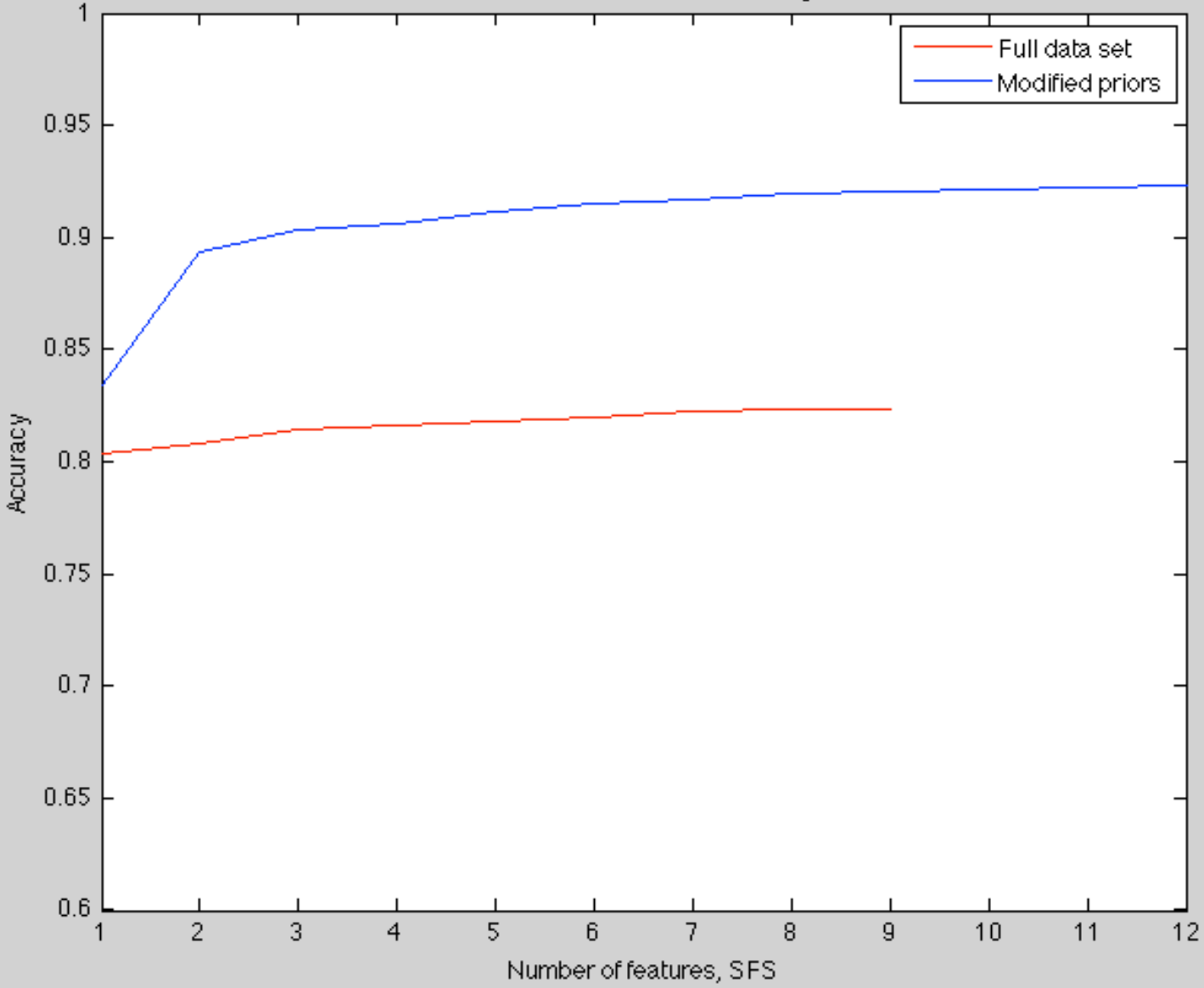
- Assumes probability density functions are normalized Gaussian distributions
- Assumes class covariances are the same
- Class means and covariances are estimated from training set

LDA

- This method was able to be run on the entire data set.
- We also ran this method on the 30,000 point dataset with the adjusted priors
- Used SFS to find feature subset that gave best results
- SBS was also tried but the method didn't eliminate very many features

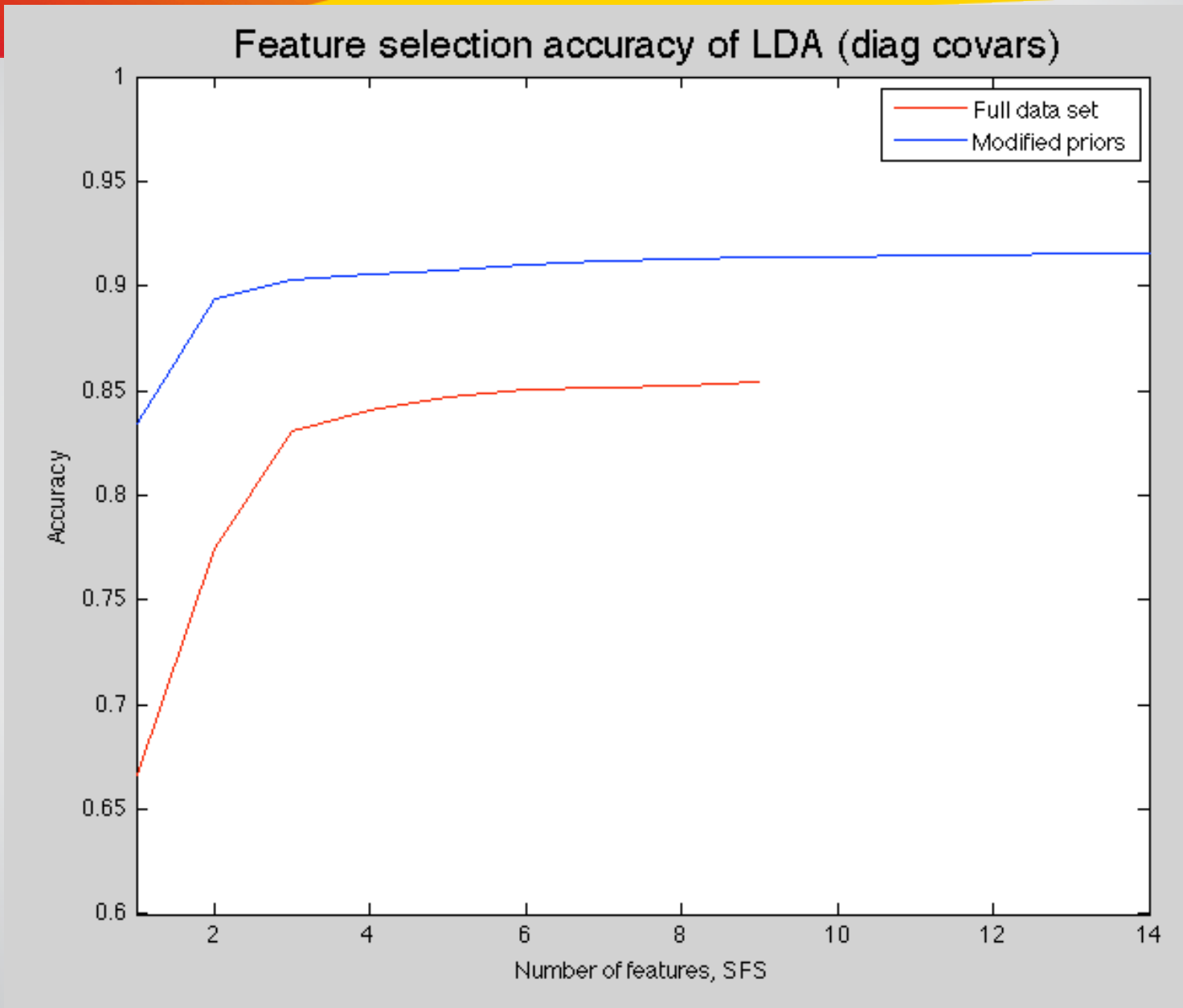
LDA

Full data set	Modified priors
Best feature subset from SFS Percentage paid off Paid amount # of borrowers # of languages # of journal entries # of commas Rate of payback Sector # of periods **Accuracy: 82.4%**	**Best feature subset from SFS** Percentage paid off Partner ID # of languages Paid amount # of borrowers Funded amount Gender # of commas Length of use Rate of payback # of journal entries Sector **Accuracy: 92.3%**


Feature selection accuracy of LDA
Full data set
Modified priors
Accuracy
Number of features, SFS
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
1
2
3
4
5
6
7
8
9
10
11
12

LDA w/ diagonal covariances

Full data set	Modified priors
Best feature subset from SFS Percentage paid off # of journal entries Partner ID Paid amount # of languages Gender # of borrowers Country code Rate of payback **Accuracy: 85.4%**	**Best feature subset from SFS** Percentage paid off Partner ID # of languages Paid amount # of borrowers Gender Rate of payback # of commas # of periods Length of use # of journal entries Funded amount Length of description Sector **Accuracy: 91.6%**


Feature selection accuracy of LDA (diag covars)
Full data set
Modified priors
Accuracy
Number of features, SFS
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
2
4
6
8
10
12
14

LDA summary

- Features common to all LDA methods
  - Percentage paid off
  - Paid amount
  - # of borrowers
  - # of languages
  - # of journal entries
  - Rate of payback
- Classification accuracy improved on data set with a more even prior probability distribution

Mutual Information

- Used as a feature selection method (to compare with sequential forward selection)
- Mutual information can be tricky in that certain features may be more “informative” when combined with other features
- Calculated mutual information of single, pairs, and triplets of features

Mutual Information

Top 8 Features:

1. Rate of repayment
2. Percentage paid back
3. Paid amount
4. Partner ID
5. Country
6. Number of borrowers
7. Funded amount
8. Length of English description

Calculated:

- $I(X_i, Y)$
- $I(X_i, X_j, Y)$
- $I(X_i, X_j, X_k, Y)$

Red indicates features that were common to **all** LDA methods (using SFS) and mutual information

Support Vector Machines

- Linear
  - C=1
  - Accuracy: 86.67%
- Radial Basis Kernel
  - C=1000
  - G=0.00001
  - Accuracy: 93.36%
- 10-fold cross validation
- 90% train, 10% test
- # data points >> # features
- Mapping data to a higher dimensional space via a nonlinear kernel improves classification

Support Vector Machines

$$\text{minimize } \|\underline{\theta}\|^2 /2 + C \sum_{t=1}^{n} \xi_t$$

$$\text{subject to } y_t(\underline{\theta}^T \underline{\phi}(x_t) + \theta_0) \geq 1 - \xi_t \qquad t = 1, ..., n$$

- Support Vector Machines aim to identify the maximum margin classifier
- The value *C* in the equation above corresponds to the cost of allowing some points in the training data to be misclassified with the aim of achieving a wider margin
- The equations above are solved using quadratic programming algorithms

Support Vector Machines

- The identity and number of support vectors are dependent on the kernel choice
  - Linear kernel; # SVs: 5997
  - Radial basis kernel, C = 1000; # SVs: 5502
- … as well as the value of the cost C in the regularization term
  - Radial basis kernel C =100; # SVs: 5531
  - Radial basis kernel C =10; #SVs: 6420

K-means

- Aims to minimize within-cluster variance:

$$\underset{\mathbf{Z}}{argmin} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in Z_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

- Estimated K by minimizing Bayesian Information Criterion (BIC)
  - Used K = 20
- Accuracy: 71.11%

K-means

- Clustering depends on initial choice of centroids
  - 5 different initial positions; chose best clustering
- K-mean uses Euclidean distance to determine within-cluster variance
  - Important to set the mean and variance to zero and one, respectively, for the features
  - Important for all algorithms; but particularly important for K-mean since it uses distance as a metric

Why BIC?

- Choosing a good *K* essentially becomes a model selection problem:
  - Comparing "models" with different numbers of clusters
- As *K* increases, within-cluster variance decreases
  - May be misinterpreted as a better fit to data
- BIC addresses the problem of over fitting
- BIC includes a penalty term for the number of parameters in the model (regularization)

BIC

$$BIC_k = -2MLL_k + d_k log(n)$$

Where

- BIC_k is the BIC value for the model M_k
- MLL_k is its maximum log likelihood
- d_k is the number of free parameters to be estimated
- n is the number of data points

Minimize BIC_k to find the optimal value for K

BIC: Obtaining MLL_K

- K-means is a hard classifier
- It generates disjoint clusters so that each data point can only belong to one cluster only
- Therefore, it does not produce likelihoods
- To obtain likelihoods, we will interpret the clusters generated as Gaussian mixtures

BIC: Calculating MLL_K

$$MLL_k = \sum_{t=1}^{n} log \sum_{z=1}^{k} q(z) N(\mathbf{x}_t; \boldsymbol{\mu}_z, \sigma \mathbf{I})$$

Where

- $\mathbf{x}_t$ is the observed parameter vector
- $\boldsymbol{\mu}_z$ is the vector containing the cluster centroids
- $\sigma \mathbf{I}$ is the isotropic variance matrix for all clusters
- *q(z)* is the weight of each cluster

K-means Pseudocode: Obtaining *K*

FOR *K*=1:30

FOR *iteration*= 1:5

Perform K-means

END

Accept K-means with the best clustering (evaluated based on lower within-cluster variance)

$$optimal\_K = \underset{K}{argmin}(BIC)$$

END

K-means Pseudocode: Training

Run K-means to cluster data into *optimal_K* clusters

FOR *Cluster*= 1:*optimal_K*

FOR *label*= 1:4

$$\Delta_{probability}(Label) = P(Label\_in\_cluster) - P(Label\_in\_dataset)$$

END

$$Cluster\_label = \underset{label}{argmax}(\Delta_{probability})$$

END

K-means

- The labels didn't have an even distribution (13%, 13%, 37%, 37%)
- Therefore, we simply couldn't choose the mode of each cluster as its label
- The performance of K-means was determined using cluster labels that were chosen by the algorithm presented in the previous slide

Gentle AdaBoost

- Adaptive: subsequent classifiers are tweaked in favor of points misclassified by previous classifiers
- Boosting: combines simple base classifiers to produce a stronger ensemble

Gentle AdaBoost can better handle outliers than classic AdaBoost

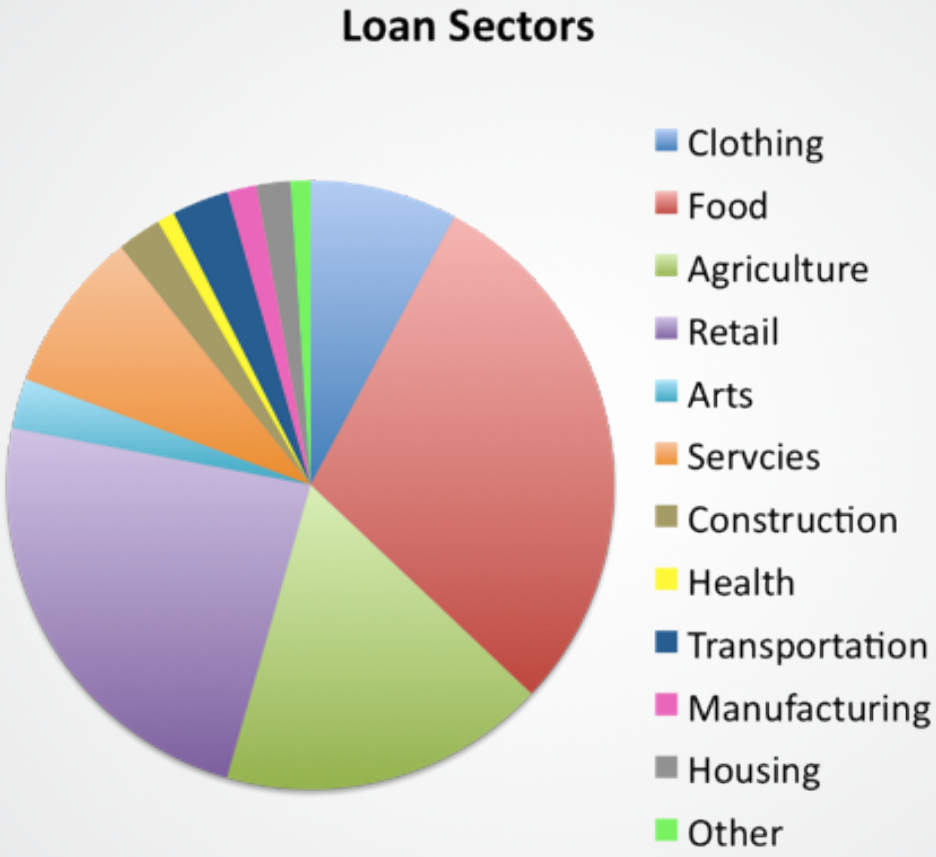
Gentle AdaBoost

- Used decision stumps as classifiers
  - 50 stumps; accuracy: 75.73%
  - 100 stumps; accuracy: 80.20%
  - 200 stumps; accuracy: 81.83%

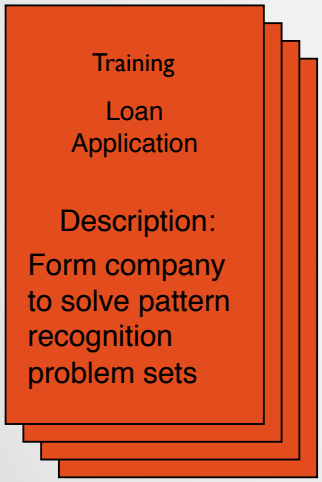
- Accuracy tends to increase as more stumps are added into the ensemble

GOAL

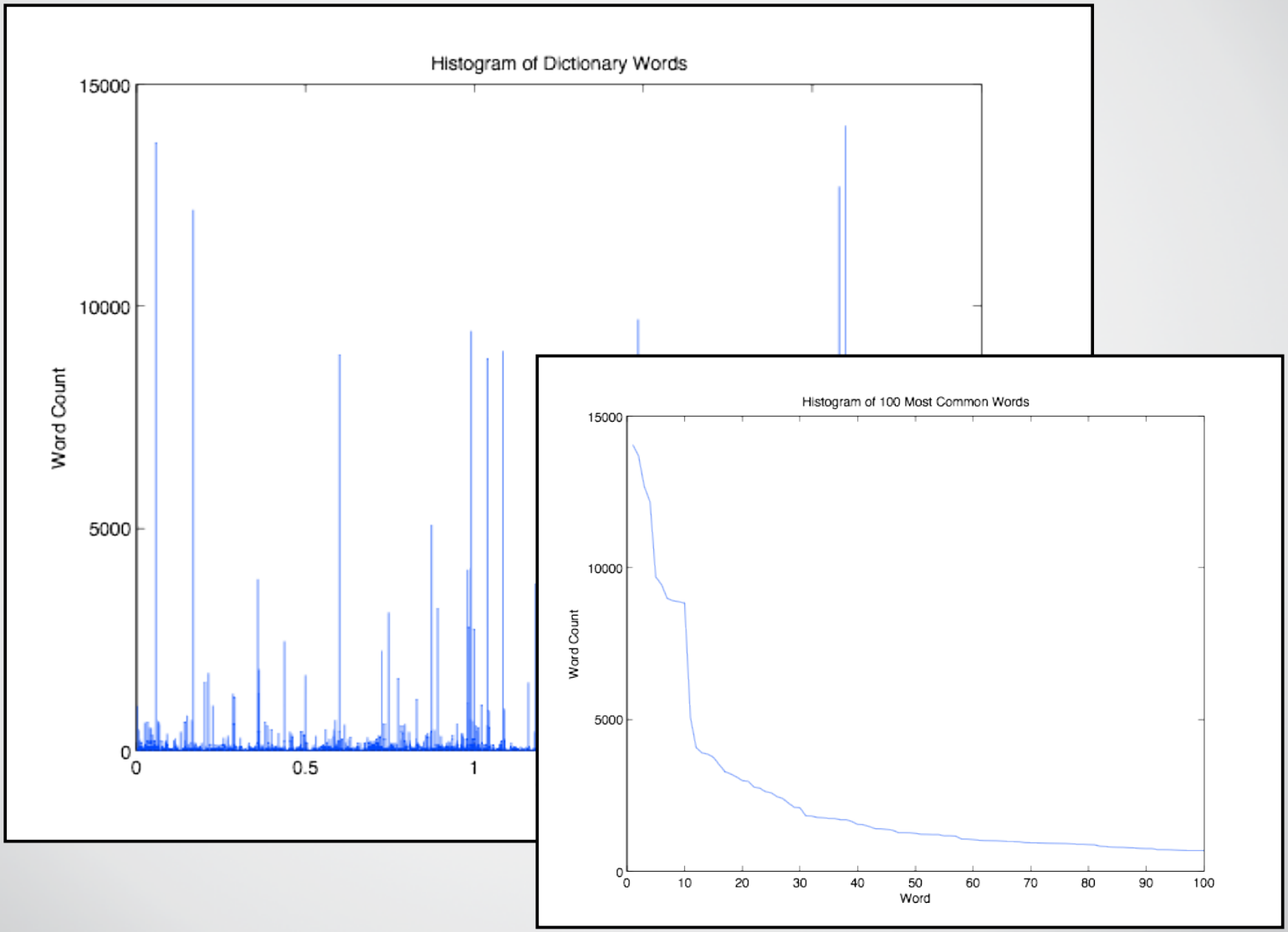
Categorize loan sector using description text

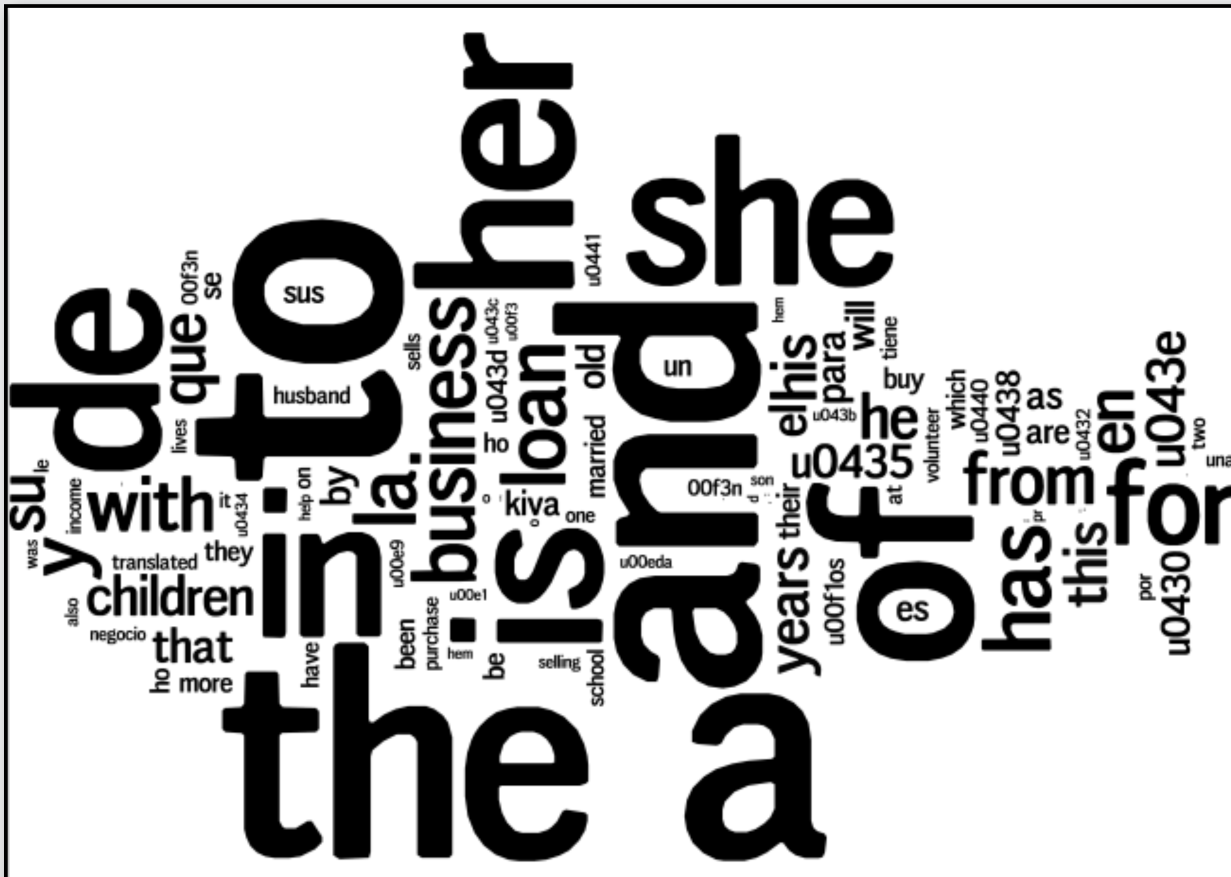

Loan Sectors
Clothing
Food
Agriculture
Retail
Arts
Servcies
Construction
Health
Transportation
Manufacturing
Housing
Other

METHODOLOGY - TRAINING

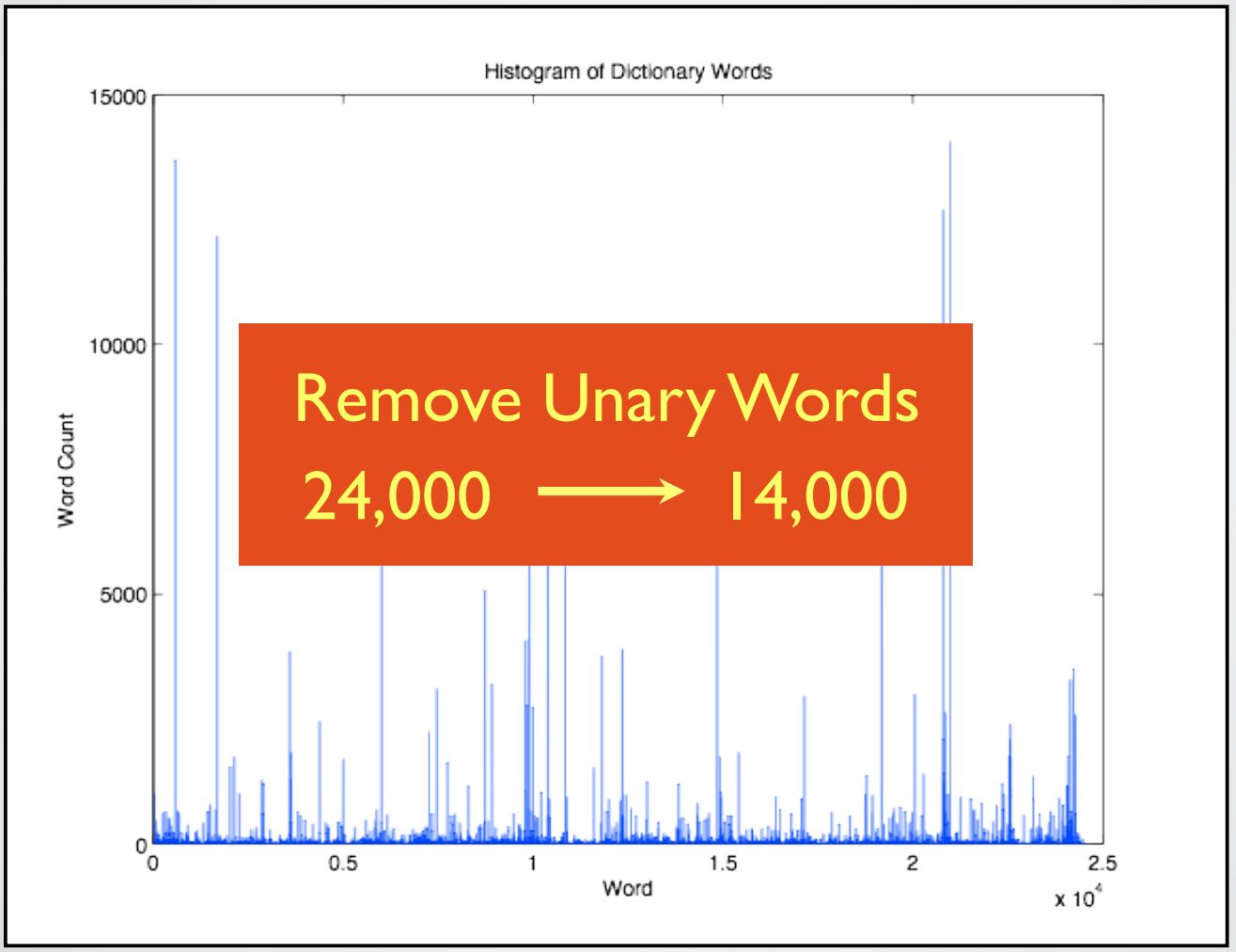


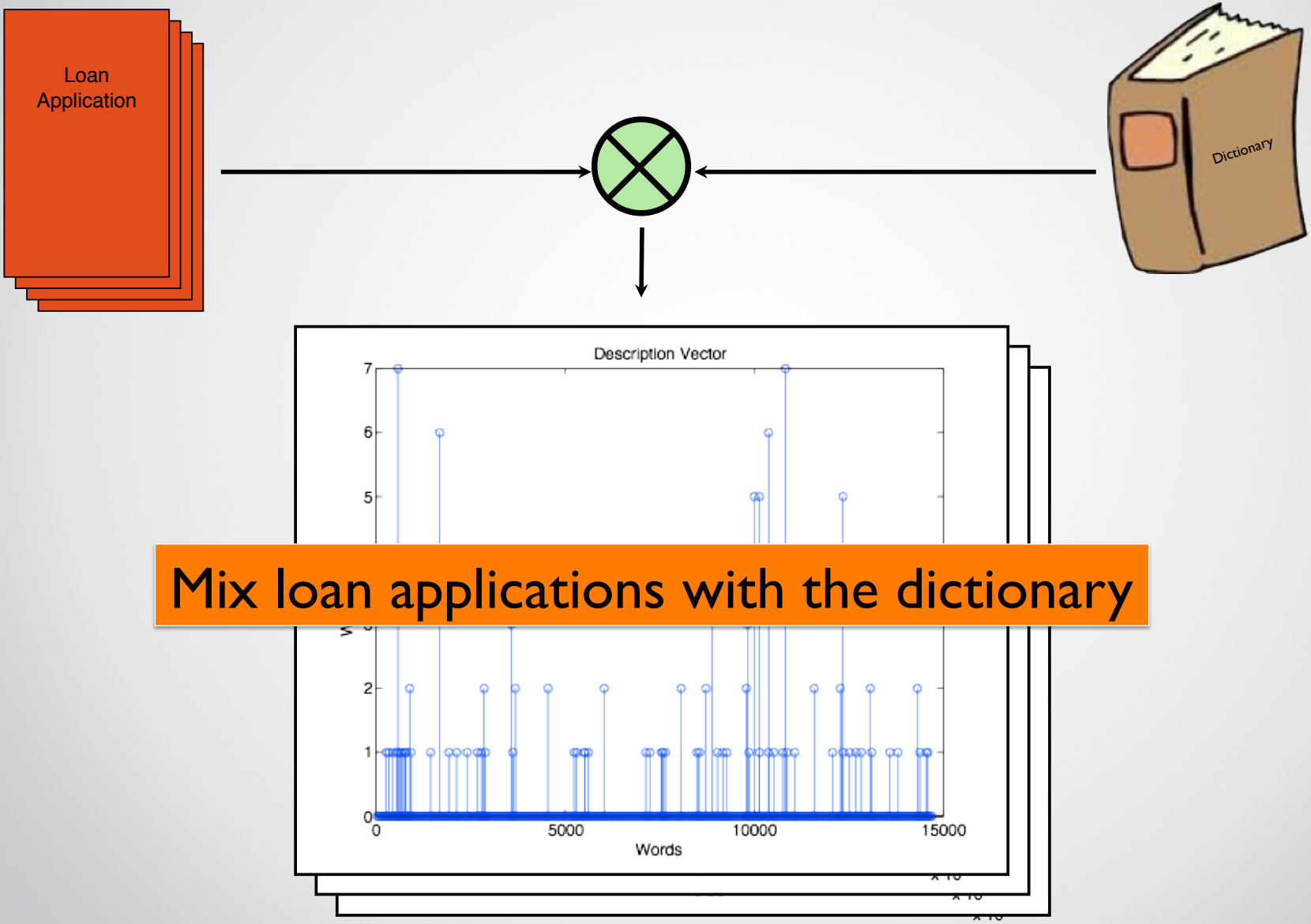
Form a (large) dictionary from the loans


Histogram of Dictionary Words
Word Count
15000
10000
5000
0
0
0.5
1
Histogram of 100 Most Common Words
Word Count
15000
10000
5000
0
0
10
20
30
40
50
60
70
80
90
100
Word

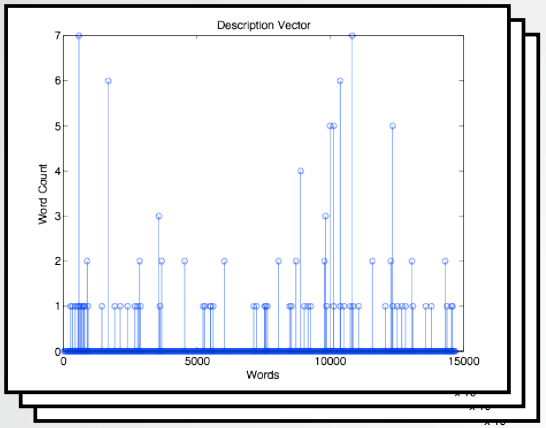


100 Most Common Word Cloud


Histogram of Dictionary Words
15000
10000
5000
0
Word Count
0
0.5
1
1.5
2
2.5
Word
x 10^4
Remove Unary Words
24,000 ⟶ 14,000

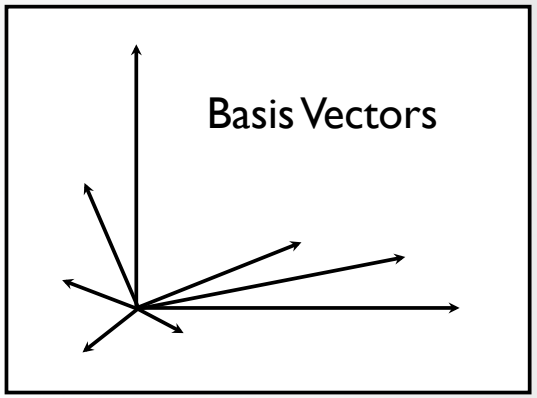


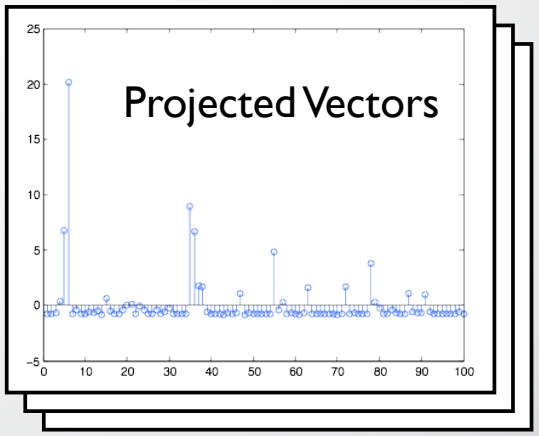
Mix loan applications with the dictionary

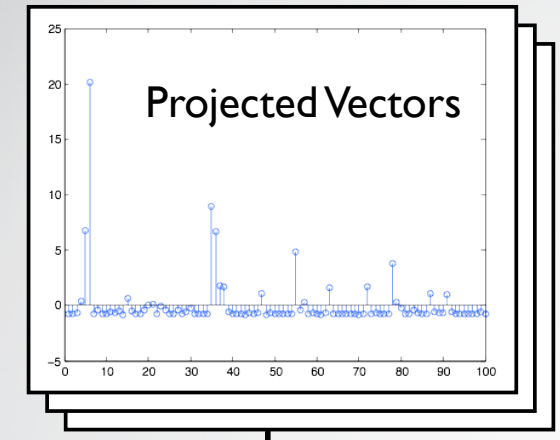


Reduce Dimensionality

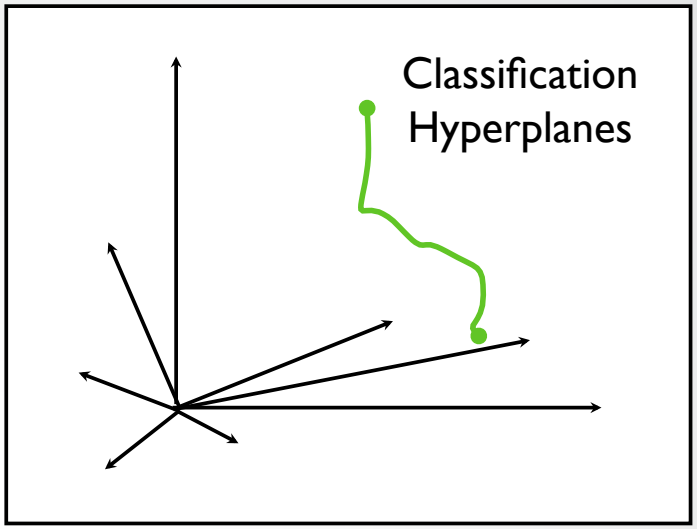
In this case PCA

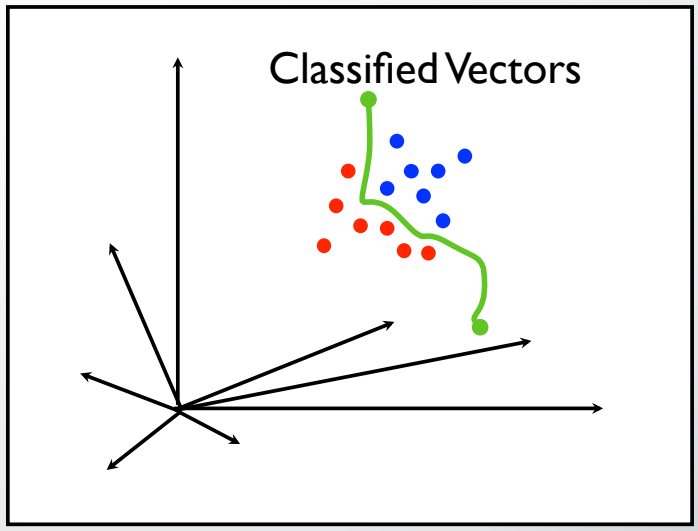


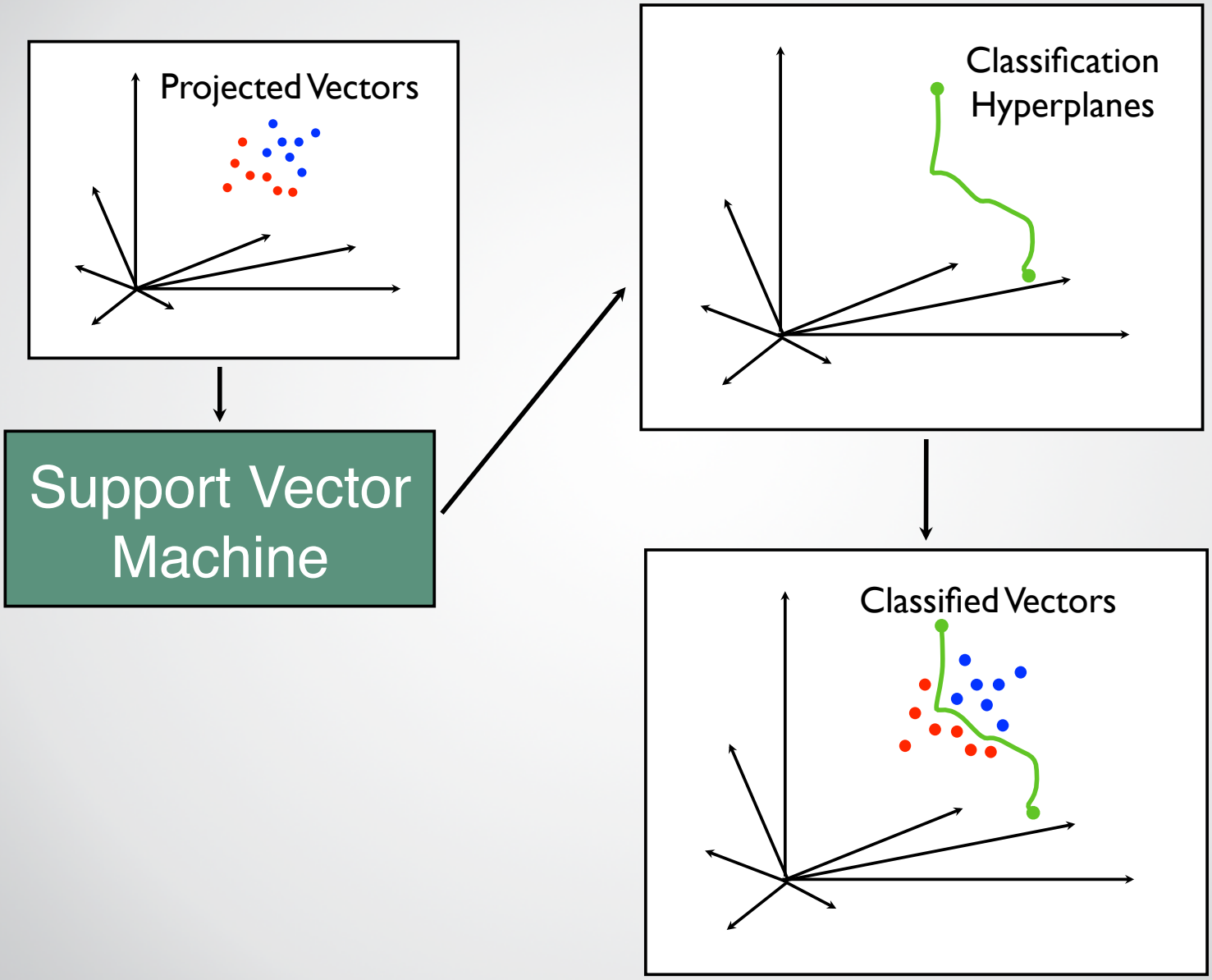




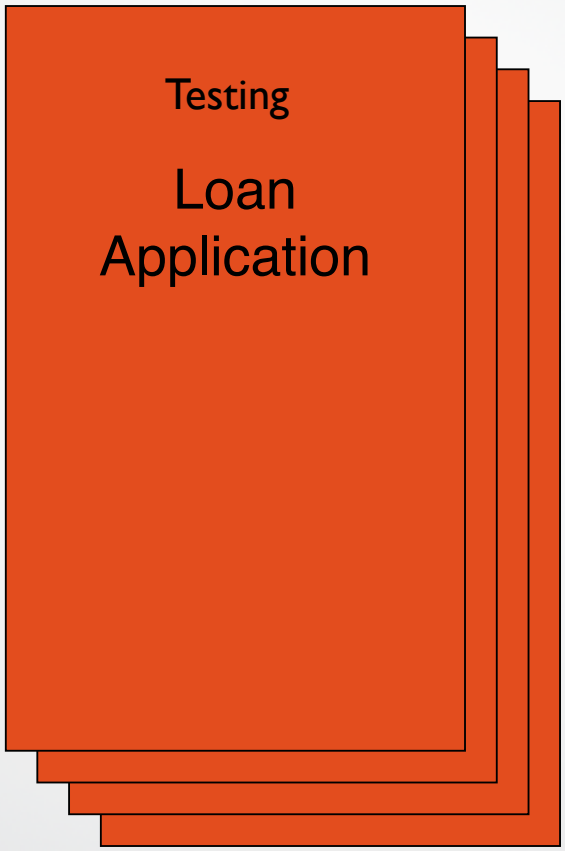
Support Vector Machine

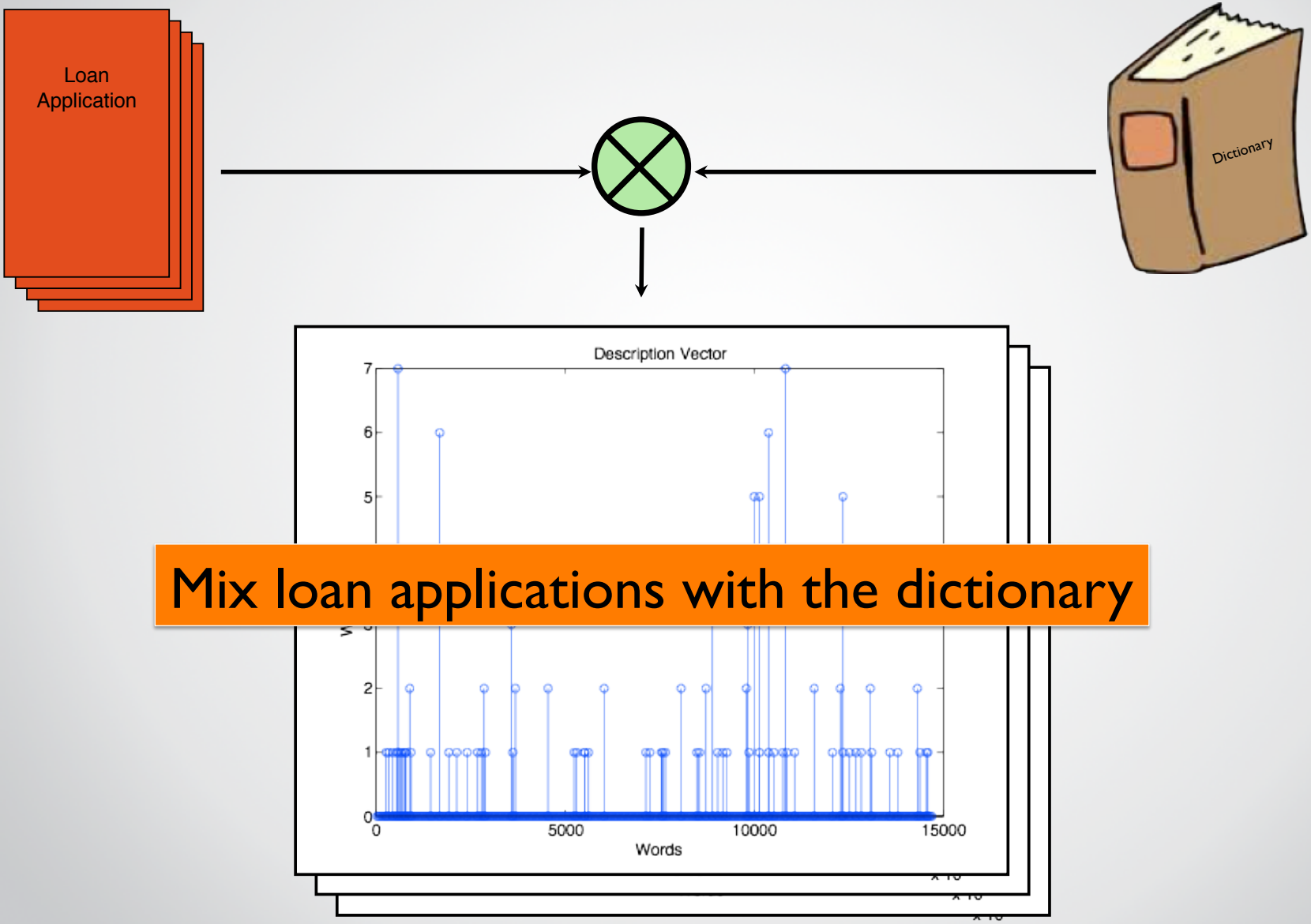




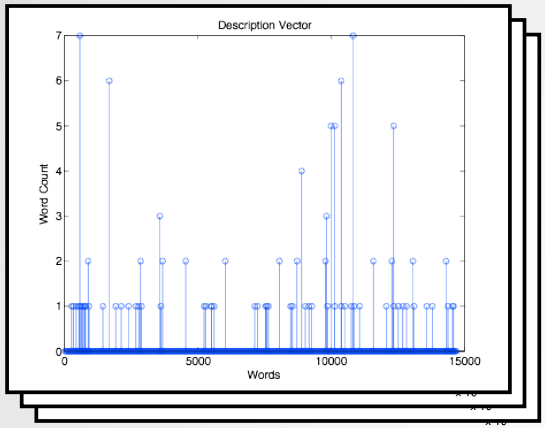

Projected Vectors
Support Vector Machine
Classification Hyperplanes
Classified Vectors

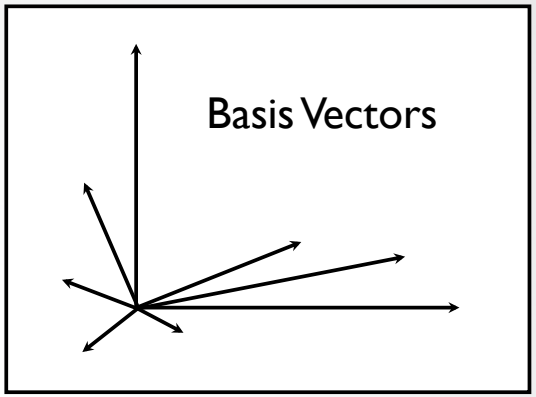
METHODOLOGY - TESTING



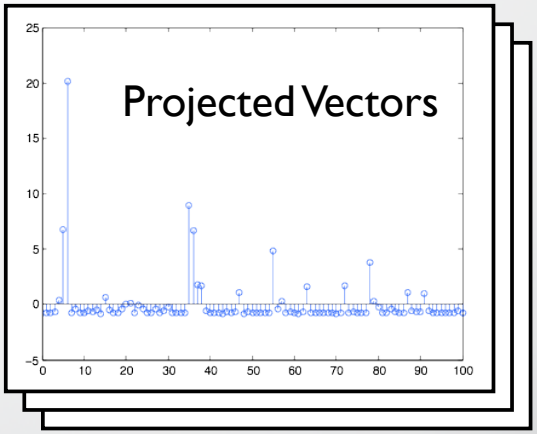


Mix loan applications with the dictionary

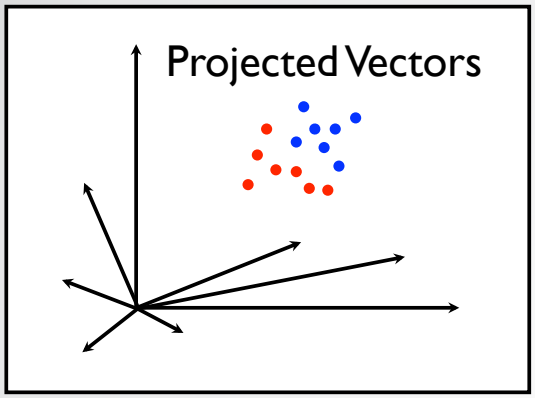


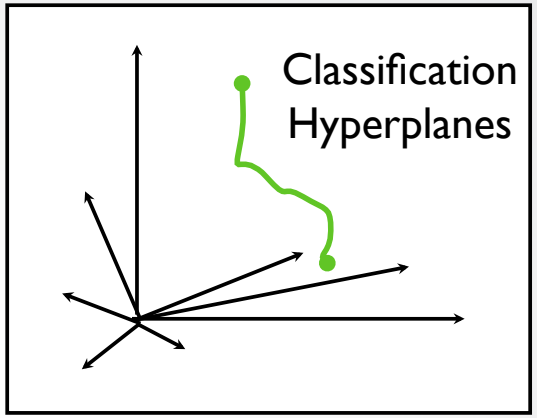


Reduce Dimensionality

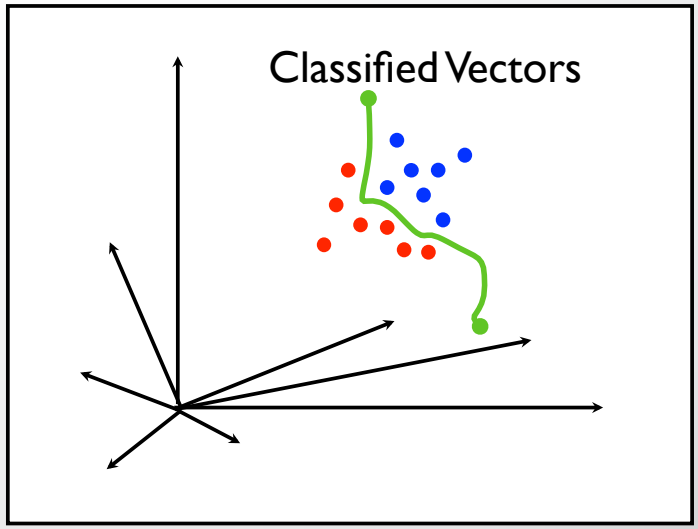


Map vectors to previously compued basis vectors





Support Vector Machine



Classify with previous hyperplanes

RESULTS

Dimensionality Reduction	SVM Kernel	Accuracy
PCA, 500 Components	Linear	61.3%
PCA, 500 Components	Epsilon SVR	74.9%
PCA, 1000 Components	Epsilon SVR	76.2%

Conclusions: Performance

Accuracy:

- KNN: 49.2%
- K-means: 71.11%
- AdaBoost: 81.83%
- Linear Kernel SVD: 86.67%
- LDA with diagonal covariance: 91.6%
- LDA: 92.3%
- Radial Basis Kernel SVM: 93.36%

Conclusions: Best Method

- Radial Basis Kernel SVM performed the best
- Since # data points >> # features, mapping data to a higher dimensional space via a nonlinear kernel improves classification
- It would be interesting to try other nonlinear methods to see if they perform as well

Conclusions: Worst Methods

- Distance-based metrics, KNN & K-means, performed the worst
- This could be due to the fact that Euclidean distance becomes a worse metric of distance as the number of features increases
- LDA and AdaBoost, which are linear methods, performed significantly better than KNN & K-means
  - Therefore, we know that the data is indeed linearly separable
  - KNN & K-means' relatively poor performance could not be due to the data being potentially nonlinear

Conclusions

- Modern pattern recognition methods can be successfully applied to the analysis to loan microfinance data
- Hands on application of these techniques expose subtleties not apparent to the casual scholar
- With large datasets some algorithms become computationally handicapped

Future Work

- Run all methods on full data set
  - Limited by computing power and time
- Do bag-of-words methods on whole dictionary
- Compare loans from different sectors/countries
- Create UI to help borrowers solidify their business plans, possibly help their chances of repayment