

Natural Language Processing for Smart Baseball Scouting

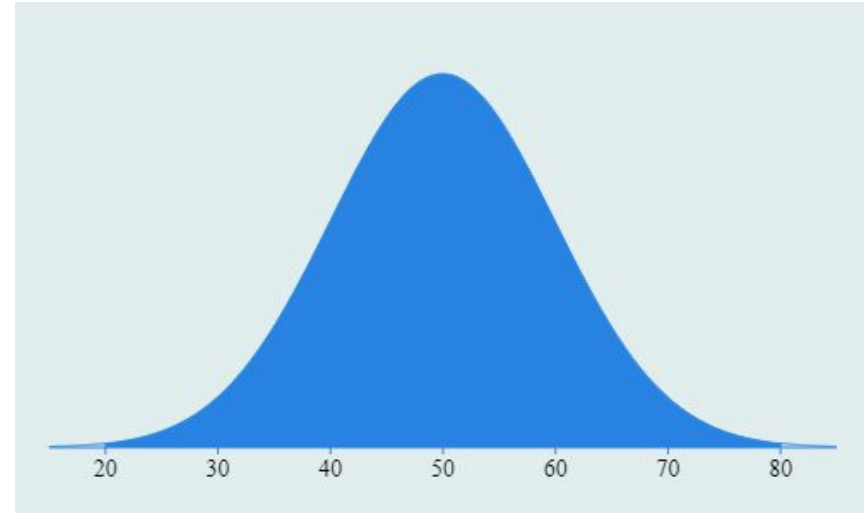
Tyra Pitts, Stephen Cha, and Daniel Crawford



Background

From a Major League Baseball team, we are tasked with

- Baseball players are rated on 20-80 scale
- 20-80 scale FOLLOWS a normal distribution, $X \sim N(50, 10)$
- Represents how good the player is, or
- Expectations for their future skills



Problem Statement

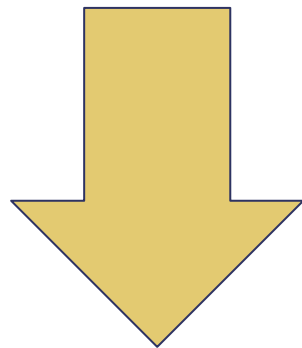
Based on an analysis of comments regarding a player, can we predict the score they will be given?

Can we train an Algorithm to pick up on the patterns of commentating to accurately score players?



WORDS

(INPUT)



NUMBERS

(OUTPUT)

Methodology Overview

**Data
Cleaning**



NLP



**Machine
Learning**

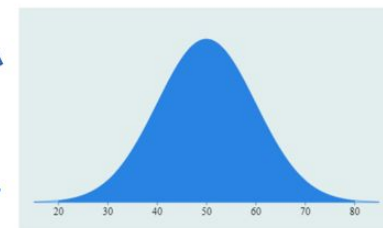
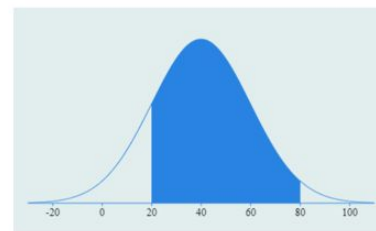
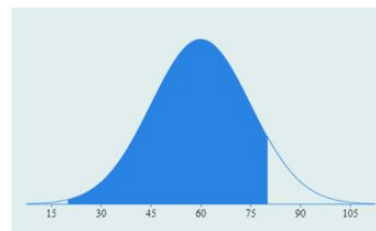
Data Normalization

Apart from normal cleaning...

30% are D,	30% are 40,	$D \rightarrow 40,$
40% are C,	40% are 50,	$C \rightarrow 50,$
20% are B,	20% are 60,	$B \rightarrow 60,$
10% are A	10% are 70	$A \rightarrow 70$

Using R, calculate cumulative distributions and map letter grades to numeric

Normalize each
source's comments to
follow $N(50,10)$:



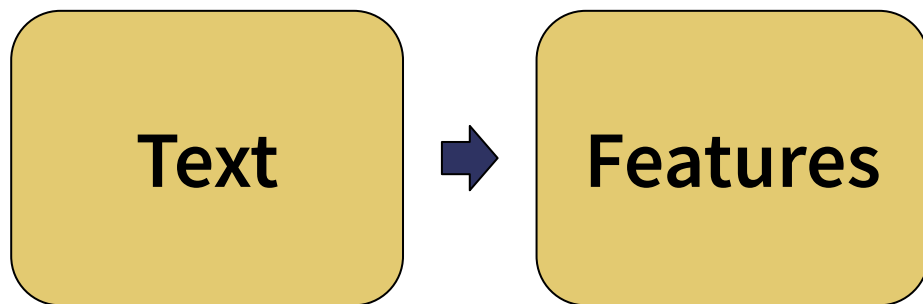
What is NLP?

- Natural Language Processing
 - Enabling computers to understand natural languages
- Applications of NLP
 - Siri, Alexa, Cortana
 - YouTube auto-caption generator
 - Text auto complete
 - Grammar/spell checker



Why Use NLP?

- We want to see if we can predict a player's grade based on a player's description
- We want to use machine learning
- To use machine learning, we need a feature vector



The Importance of a Feature Vector

- Say we want to predict a student's grade taking a Dr. Wheeler math class
- What would be good features to predict a student's grade?
- Perhaps something like:

$$\vec{x} = \begin{bmatrix} \text{Number of hours studied} \\ \text{Number of hours playing video games} \\ \text{Number of times shown up for class} \\ \text{Number of times visited Dr. Wheeler's Office Hours} \\ \text{Number of beers offered to Dr. Wheeler} \end{bmatrix}^T$$



Initial Approach

- Each unique word assigned an integer index value
- Reserve certain indices for “special” words
 - “Start”
 - “Unknown”
- Observe how index word count frequency affects grade



Bag-of-words

- Document
 - A unit of text
 - Comment, sentence, paragraph, etc.
- Bag-of-words
 - Vector representation of a document
 - Presence, frequency, TF-IDF



Presence Bag-of-words

- For example, “bears question dream bears”

“bears”	“beets”	“question”	“battlestar”	“dream”	“galactica”
True	False	True	False	True	False

Frequency Bag-of-words

- For example, “bears question dream bears”

“bears”	“beets”	“question”	“battlestar”	“dream”	“galactica”
2	0	1	0	1	0

TF-IDF Bag-of-words

- For example, “bears question dream bears”

“bears”	“beets”	“question”	“battlestar”	“dream”	“galactica”
7.23	0.02	2.39	0.19	2.14	0.08



TF-IDF

- TF-IDF (Term Frequency-Inverse Document Frequency)
- In a nutshell:
 - For each word, assign a weight
 - Weight increases as a term appears more often in a given document
 - Weight decreases as a term appears across multiple documents

$$w(t, c) = tf(t, c) \times \ln \left(\frac{N}{1 + df(t)} \right)$$

One Consideration

- Word succession order matters
- Certain word groupings have different connotations, changing effect on rating

Word Order	Meaning
The <u>cat</u> sat on the <u>girl</u> .	
The <u>girl</u> sat on the <u>cat</u> .	

Sentiment Analysis

Average or below	Above Average
"this player sucks"	"this player is bonkers amazing!"

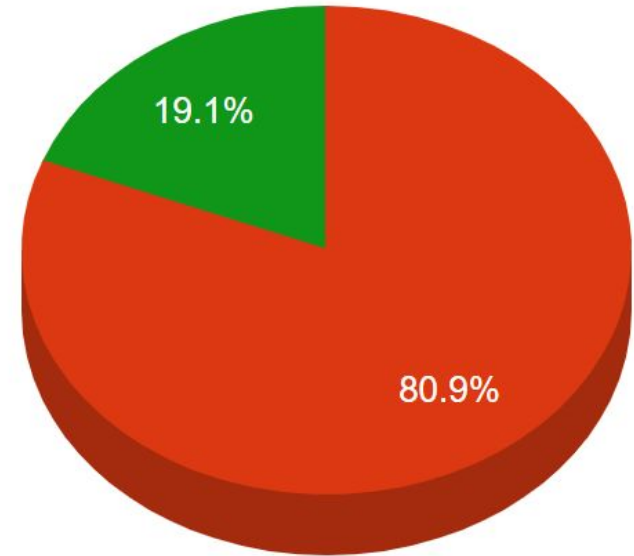
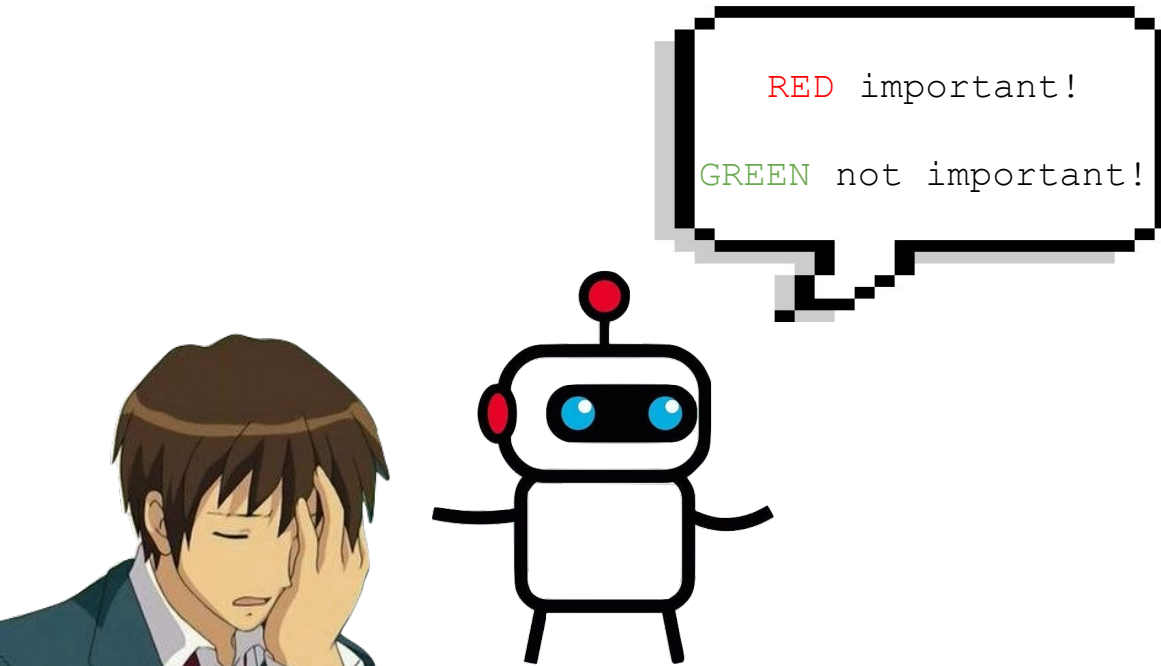


Stop Words

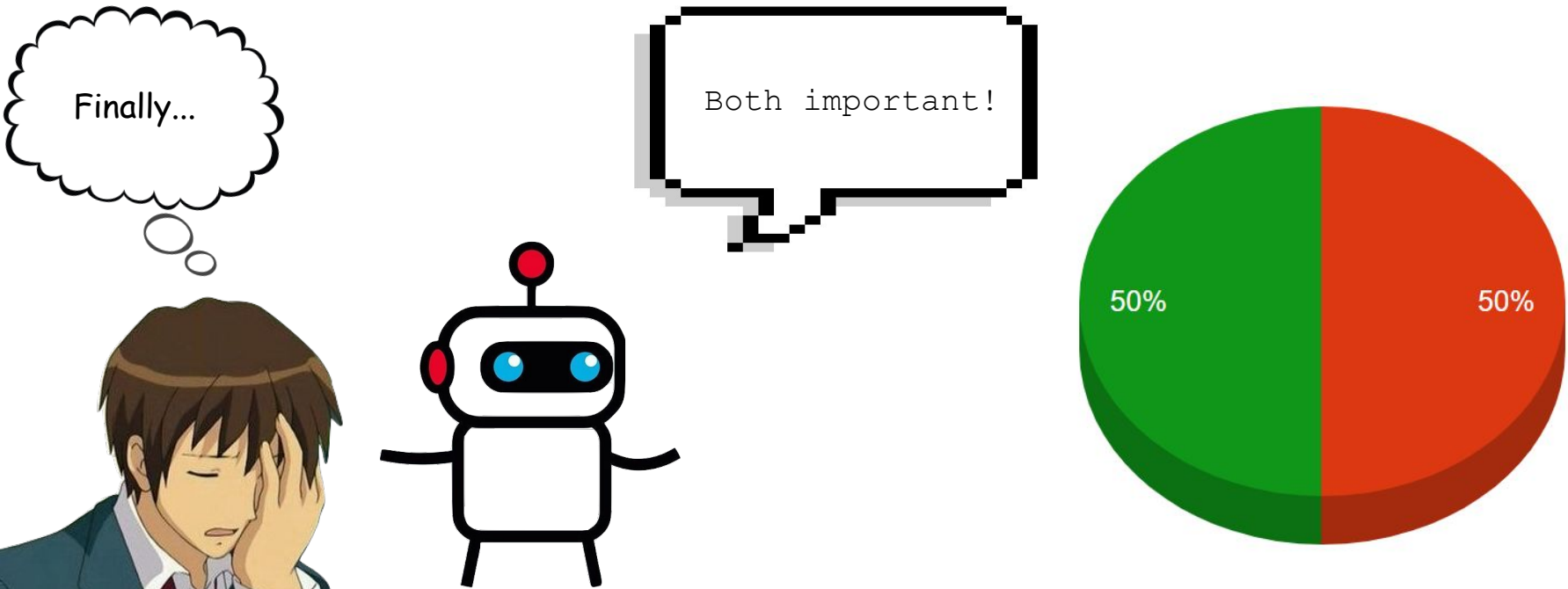
- Stop words: words that are filtered out
 - Ex) the, in, is, which, etc.

Raw Text	After Stop Words Removal
“the duck swam in the lake”	“duck swam lake”

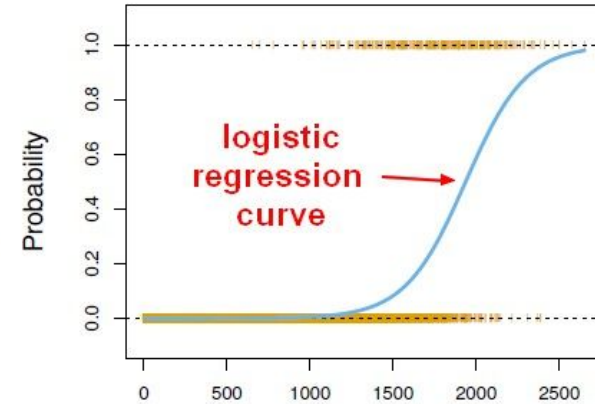
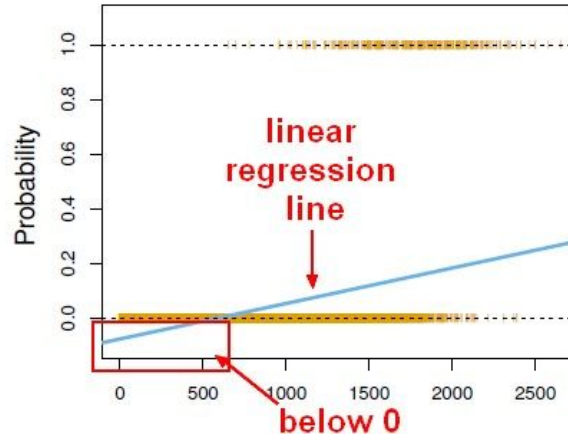
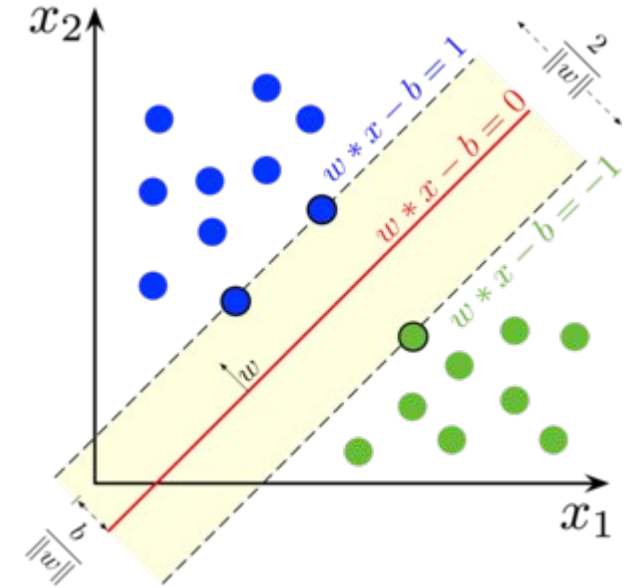
Class Imbalance



Class Imbalance



The Machine Learning “Engine”

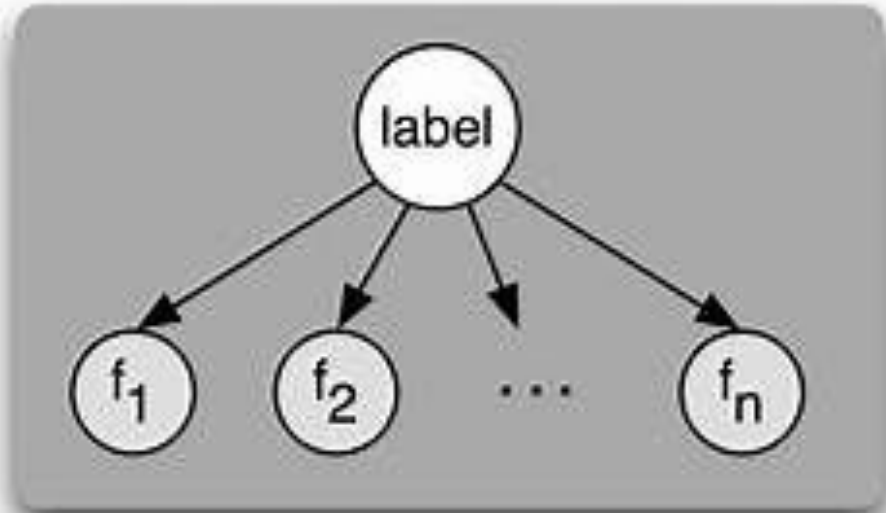


$$\arg \max_{\beta} : \log \left\{ \prod_{i=1}^n P(y_i|x_i)^{y_i} (1 - P(y_i|x_i))^{(1-y_i)} \right\}$$

LONG GREEN THIN



More Algorithms: Naive Bayes



$$p(C_k | \mathbf{x}) = \frac{p(C_k) p(\mathbf{x} | C_k)}{p(\mathbf{x})}$$

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

$$\hat{y} = \operatorname{argmax}_{k \in \{1, \dots, K\}} p(C_k) \prod_{i=1}^n p(x_i | C_k).$$

$$p(\mathbf{x} | C_k) = \prod_{i=1}^n p_{ki}^{x_i} (1 - p_{ki})^{(1-x_i)}$$

Running ML

BOW Type	ML Algorithm	Accuracy	Recall
TF-IDF	SVM	82.05%	72%
TF-IDF	Logistic regression	82.94%	73%
Presence	Bernoulli NB	82.94%	76%
Frequency	Logistic regression	78.94%	76%
TF-IDF	Complement NB	75.07%	78%
Frequency	Complement NB	83.53%	65%
Frequency	SVM	73.00%	99%

**Recall -
Proportion of True
Positives Correctly
Identified**

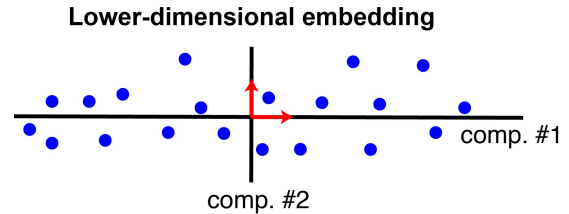
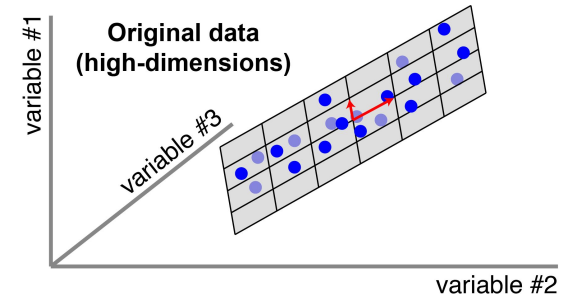
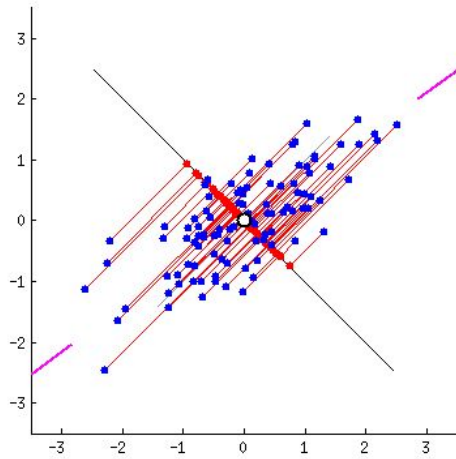
Bag-of-words Concerns

1. High dimensionality
2. Assumes words are independent
 - For example,
 - Words like “great”, “awesome”, and “fantastic” are considered completely different
 - Even though that’s not how we understand those words



Principal Component Analysis

- Projects data points onto lower dimensional subspace
- Pick subspace that minimizes orthogonal distance



Principal Component Analysis

- Let m be number of data points, n be number of features (dimensionality)
- Compute an n by n matrix

$$C = \frac{1}{m} \sum_{i=1}^n (x^{(i)})(x^{(i)})^T$$

- Compute singular value decomposition

$$C = U\Sigma V^T$$



Singular Value Decomposition

- We're concerned with only

$$U = \begin{bmatrix} | & | & \dots & | \\ u^{(1)} & u^{(2)} & \dots & u^{(n)} \\ | & | & & | \end{bmatrix}$$

where U is orthogonal and $u^{(i)} \in \mathbb{R}^n$

Projecting onto lower dimensional subspace

Pick a k such that $k < n$.

$$z^{(i)} = U_r^T x^{(i)} = \begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \dots & u^{(k)} \\ | & | & & | \end{bmatrix}^T \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^k$$

Note that...

$$\begin{aligned}
 z^{(i)} &= \begin{bmatrix} u_1^{(1)} x_1^{(i)} + u_2^{(1)} x_2^{(i)} + \cdots + u_n^{(1)} x_n^{(i)} \\ u_1^{(2)} x_1^{(i)} + u_2^{(2)} x_2^{(i)} + \cdots + u_n^{(2)} x_n^{(i)} \\ \vdots \\ u_1^{(k)} x_1^{(i)} + u_2^{(k)} x_2^{(i)} + \cdots + u_n^{(k)} x_n^{(i)} \end{bmatrix} \\
 &= \begin{bmatrix} u_1^{(1)} \\ u_1^{(2)} \\ \vdots \\ u_1^{(k)} \end{bmatrix} x_1^{(i)} + \begin{bmatrix} u_2^{(1)} \\ u_2^{(2)} \\ \vdots \\ u_2^{(k)} \end{bmatrix} x_2^{(i)} + \cdots + \begin{bmatrix} u_n^{(1)} \\ u_n^{(2)} \\ \vdots \\ u_n^{(k)} \end{bmatrix} x_n^{(i)}
 \end{aligned}$$

Why use PCA?

- Outputs new features that are linear combinations of original features
 - In our case, linear combinations of words
- An attempt to capture semantic similarities between words
- Heard about it everywhere:
 - All over the internet
 - People that know ML
 - Numerical linear algebra class



Running ML with Stemming & PCA

BOW Type	ML Algorithm	Accuracy	Recall
TF-IDF	SVM	82.79%	75%
TF-IDF	Logistic regression	84.72%	71%
Presence	Bernoulli NB	65.73%	54%
Frequency	Logistic regression	78.78%	66%
Frequency	SVM	74.48%	63%



Results Summary

With Stemming and PCA

BOW Type	ML Algorithm	Accuracy	Recall
TF-IDF	SVM	82.79%	75%
TF-IDF	Logistic regression	84.72%	71%
Presence	Bernoulli NB	65.73%	54%
Frequency	Logistic regression	78.78%	66%
Frequency	SVM	74.48%	63%

BOW Type	ML Algorithm	Accuracy	Recall
TF-IDF	SVM	82.05%	72%
TF-IDF	Logistic regression	82.94%	73%
Presence	Bernoulli NB	82.94%	76%
Frequency	Logistic regression	78.94%	76%
TF-IDF	Complement NB	75.07%	78%
Frequency	Complement NB	83.53%	65%
Frequency	SVM	73.00%	99%



What's Next?

- Further data cleaning
 - Tokenize certain words and phrases
- Obtain more data
 - Training set might be not representative
- Look into alternatives:
 - **Principal component analysis** to **latent semantic analysis**
 - **Bag-of-words** to **Word Embedding**
- Transition from predicting sentiment to predicting grade
 - Will be hard because of class imbalance
- Apply regression



Regression

- Predictive equation
- Use numerical ratings
- NOT Classification

Dependent Variable

Population Y intercept

Population Slope Coefficient

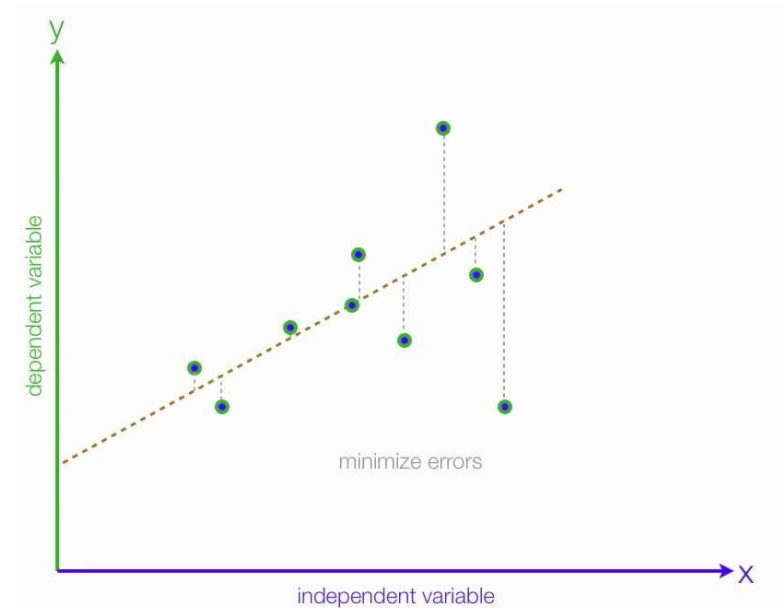
Independent Variable

Random Error term

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

Linear component

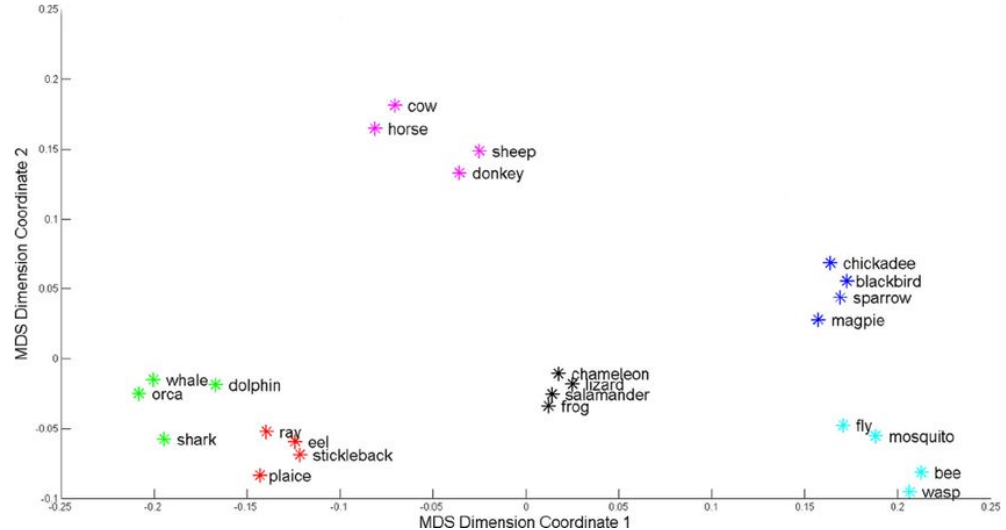
Random Error component



RSE: 7.247
R²: ~0.4336
P-values < 2.2e-16

Latent Semantic Analysis (LSA)

- NLP dimension-reduction technique
- Word association within text corpus
- Uses Bag-of-Words model
- Occurrence Matrix



Latent Semantic Analysis (LSA)

- Occurrence Matrix of comment word counts
- Rows = unique words
- Columns = single comment
- Tf-idf weight of important words

		Documents					
		doc_1	doc_2	doc_3	doc_4	doc_5	...
Terms	I	1	0	3	5	9	...
	love	2	1	5	3	4	...
	burger	3	6	4	2	1	
	...	⋮	⋮	⋮	⋮	⋮	...

Singular Value Decomposition (SVD)

- Singular Value Decomposition minimizes number of rows
 - 2 columns normalized
 - Dot product
 - 1 represents similar comments
 - 0 represents differing comments

$$M=U\Sigma V^*$$

U is left singular matrix; **Σ** is diagonal matrix; **V^*** is transposed right singular matrix

MM^*

M^*M



Very Special Thanks!

Joe Datz, our liaison and point man for the project

The Pittsburgh Pirates, for providing us with a project and data

Dan Fox, for approval of project

Nathan Ong, for being an excellent resource in many facets of the project



Special Thanks!

Pitt Mathematics Department, for providing us this opportunity and funding

Professor Jeff Wheeler, for his unwavering support and motivation



THANK YOU



Very Special Thanks!!!

Professor Jeff Wheeler,

for his unwavering support and motivation

University of Pittsburgh,

for graciously hosting us and supporting our class!



References

- <http://jakemdrew.files.wor>
- <https://www.quora.com/What-is-meant-by-entropy-in-machine-learning-contexts>
- [https://en.wikipedia.org/wiki/Latent Dirichlet allocation](https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation)
- https://images.search.yahoo.com/search/images;_ylt=A0PDsBnqpYJcPlsA5ixXNyoA;_ylu=X3oDMTB0N2Noc21lBGNvbG8DYmYxBHBvcwMxBHZ0aWQDBHNlYwNwaXZz?p=machine+learnign+comic&fr2=piv-web&fr=mcafee#id=2&iurl=https%3A%2F%2Fimgs.xkcd.com%2Fcomics%2Fmachine_learning.png&action=click
- [https://en.wikipedia.org/wiki/Ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning)
- <http://blog.christianperone.com/2011/09/machine-learning-text-feature-extraction-tf-idf-part-i/>
- All images open online (Google images), or generated by team member

