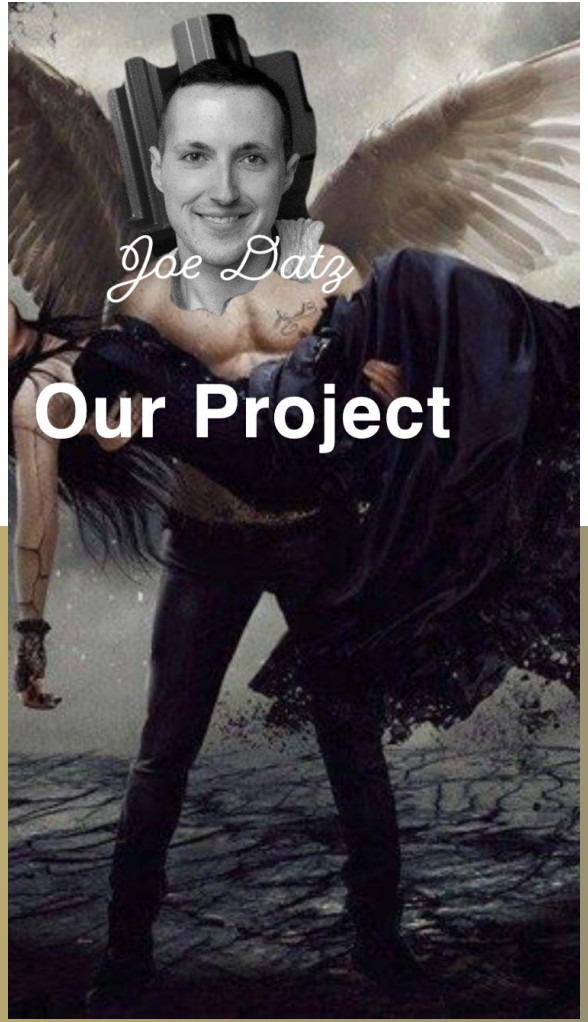
Natural Language Processing for Smart Baseball Scouting

Tyra Pitts, Stephen Cha, Daniel Crawford
Joe Datz, Jeff Wheeler

Dynamic Pricing

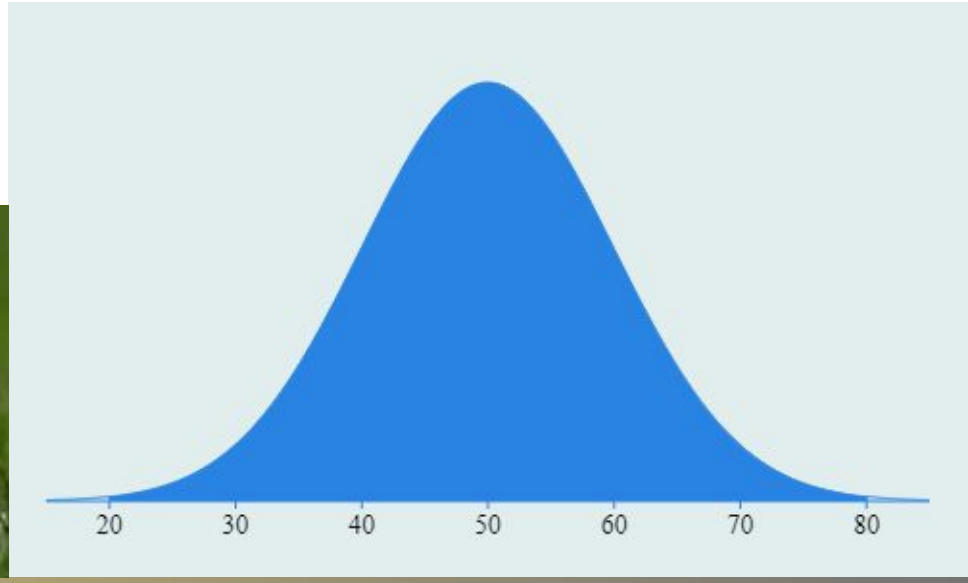
REJECTED


Tuition Efficiency
REJECTED


Joe Datz
Our Project

Background

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



Problem Statement

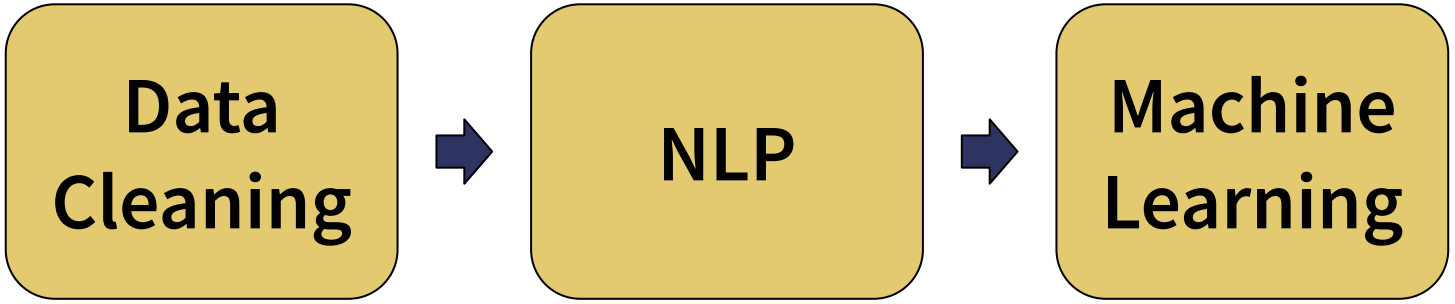
Based on an analysis of comments regarding a player, can we predict the score they will be given?

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

WORDS

NUMBERS

Methodology Overview



Data Cleaning

Apart from normal cleaning…

NUMERIC	LETTER
35	C-
40	C
45	C+
50	B-
55	B
60	B+
65	A-

30% are D,
40% are C,
20% are B,
10% are A

30% are 40,
40% are 50,
20% are 60,
10% are 70

D → 40,
C → 50,
B → 60,
A → 70

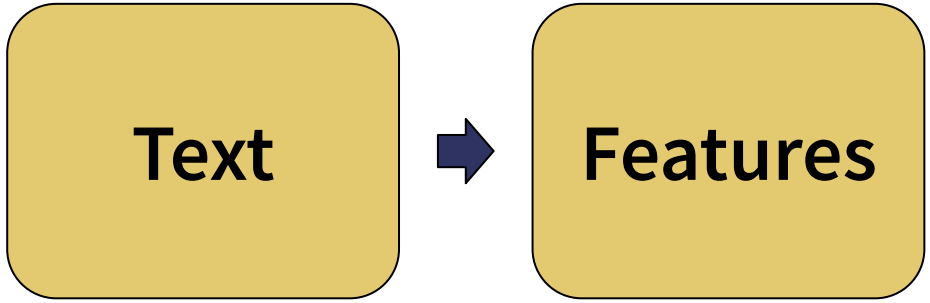
Using R, calculate cumulative distributions and map letter grades to numeric

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 critic's comment
- 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 parameters to predict a student's grade?
- Perhaps something like:

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

NLP for Text Classification

- Movie Reviews (text data)
- Each word assigned an integer index value
  - Python dictionary
- Reserve certain indices for “special” words
  - “Start”
  - “Unknown”
- Machine Learning later associates word occurrence frequency to “good” or “bad” movie score

Feature Extraction Using NLP

- Document: a unit of text (comment, review, sentence, paragraph, etc.)
- Bag-of-words (e.g. presence, frequency, weight)
- For example, “bears question dream bears”

TYPE	“bears”	“beets”	“question”	“battlestar”	“dream”	“galactica”
PRESENCE	True	False	True	False	True	False
FREQUENCY	2	0	1	0	1	0
WEIGHT	7.23	0.02	2.39	0.19	2.14	0.08

How to Compute Weights

- TF-IDF (Term Frequency-Inverse Document Frequency)
- Converts a string of text to a numeric feature vector
- In a nutshell:
  - For each word, assign a weight $w = tf \times idf$
  - If a word appears often in a single document, increase term frequency value
  - If a word appears often in multiple documents, decrease inverse document frequency value
  - So, “important” words that capture the essence of a document are boosted, words that show up in many documents are penalized

The “TF” in “TF-IDF”

Definition of Term Frequency	Example
$$tf(t,d) = \sum_{x \in d} fr(x,t),$$ $t =$ term $d =$ document $x =$ any word $$fr(x,t) = \begin{cases} 1, & \text{if } x = t \\ 0, & \text{otherwise} \end{cases}$$	$d_1 =$ “The sky is blue.” $d_2 =$ “The sun is bright.” $d_3 =$ “The sun in the sky is bright.” $d_4 =$ “We can see the shining sun, the bright sun.” $tf(\text{“}sun\text{”}, d_1) = 0+0+0+0 = 0$ $tf(\text{“}sun\text{”}, d_2) = 0+1+0+0 = 1$ $tf(\text{“}sun\text{”}, d_3) = 0+1+0+0+0+0+0 = 1$ $tf(\text{“}sun\text{”}, d_4) = 0+0+0+0+0+1+0+0+1 = 2$

The “IDF” in “TF-IDF”

Definition of Inverse Document Frequency	Example
$idf(t) = \ln\left(\frac{\lvert D\rvert}{1 + \lvert\{d : t \in d\}\rvert}\right)$ $t =$ term $d =$ document $\lvert D\rvert =$ number of all documents $\lvert\{d : t \in d\}\rvert =$ number of all documents with the term t in it	$d_1 =$ “The sky is blue.” $d_2 =$ “The sun is bright.” $d_3 =$ “The sun in the sky is bright.” $d_4 =$ “We can see the shining sun, the bright sun.” $idf(\text{“sun”}) = \ln\left(\frac{4}{1+3}\right) = \ln\left(\frac{4}{4}\right) = \ln(1) = 0$ $idf(\text{“sky”}) = \ln\left(\frac{4}{1+2}\right) = \ln\left(\frac{4}{3}\right) \approx 0.287682072$ $idf(\text{“blue”}) = \ln\left(\frac{4}{1+1}\right) = \ln(2) \approx 0.693147181$

TF-IDF

- To compute TF-IDF:

$$tf\text{-}idf(t,d) = tf(t,d) \times idf(t)$$

Batch Compute TF-IDF

Matrix multiplication $M_{tf\text{-}idf} = M_{tf} \times M_{idf}$

That is...

$$\begin{bmatrix} tf(t_1,d_1) & tf(t_2,d_1) & tf(t_3,d_1) & tf(t_4,d_1) \\ tf(t_1,d_2) & tf(t_2,d_2) & tf(t_3,d_2) & tf(t_4,d_2) \end{bmatrix} \times \begin{bmatrix} idf(t_1) & 0 & 0 & 0 \\ 0 & idf(t_2) & 0 & 0 \\ 0 & 0 & idf(t_3) & 0 \\ 0 & 0 & 0 & idf(t_4) \end{bmatrix}$$

$$= \begin{bmatrix} tf(t_1,d_1) \times idf(t_1) & tf(t_2,d_1) \times idf(t_2) & tf(t_3,d_1) \times idf(t_3) & tf(t_4,d_1) \times idf(t_4) \\ tf(t_1,d_2) \times idf(t_1) & tf(t_2,d_2) \times idf(t_2) & tf(t_3,d_2) \times idf(t_3) & tf(t_4,d_2) \times idf(t_4) \end{bmatrix}$$

Feature Vector Achieved!

- The columns of $M_{\text{tf-idf}}$ correspond to the features
- The rows of $M_{\text{tf-idf}}$ correspond to the comments
- The matrix $M_{\text{tf-idf}}$ is the input for a machine learning algorithm

$$M_{tf\text{-}idf} = \begin{bmatrix} tf(t_1, d_1) \times idf(t_1) & tf(t_2, d_1) \times idf(t_2) & tf(t_3, d_1) \times idf(t_3) & tf(t_4, d_1) \times idf(t_4) \\ tf(t_1, d_2) \times idf(t_1) & tf(t_2, d_2) \times idf(t_2) & tf(t_3, d_2) \times idf(t_3) & tf(t_4, d_2) \times idf(t_4) \end{bmatrix}$$

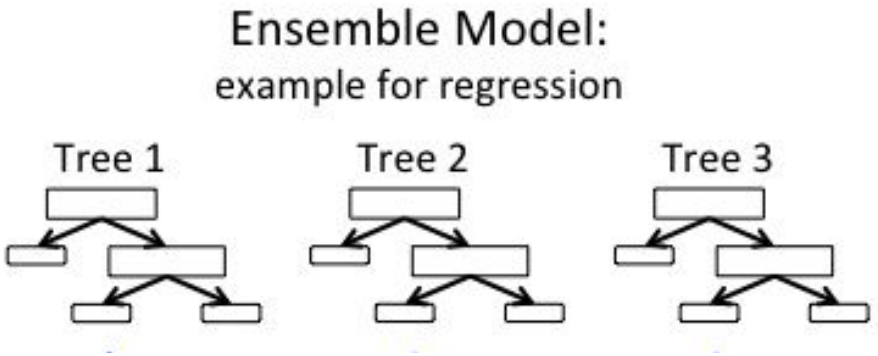
Machine Learning Algorithms

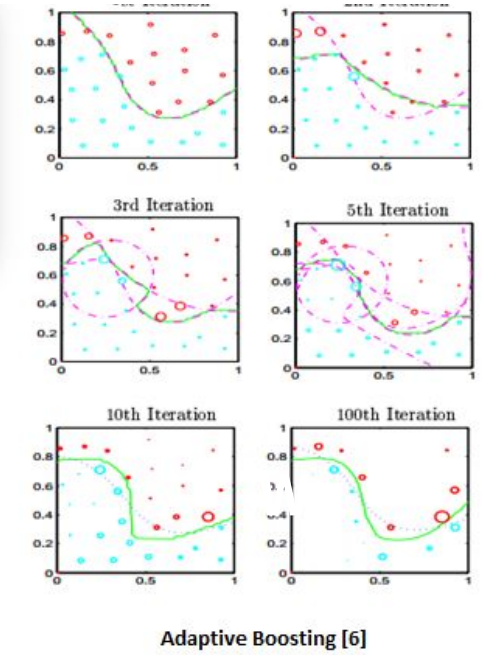
$$H(X) = -\sum_{i=1}^{n} P(x_i) log_2 P(x_i)$$

Maximum Entropy - “Least Informative”, Measure of Randomness

Support Vector Machine - Binary Classifier

Ensemble Methods - Boosting, Decision Forests, Bagging





Adaptive Boosting [6]

Conclusion & Extensions

- Normalize & clean data, increase amount of data
- Convert text to numeric values for Machine Learning
- Extract features for “important” words
- Code Python Machine Learning
- Parameters that increase or lower player’s rating
- Explore Appropriate Machine Learning Algorithms
- Configure appropriately chosen Machine Learning Algorithm

Special Thanks

Joe Datz

“Unnamed” Sports Team

Nathan Ong

Pitt Mathematics Department

Jeffrey Wheeler

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://images.search.yahoo.com/search/images;_ylt=A0PDsBnqpYJcPIsA5ixXNyoA;_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
- http://blog.christianperone.com/2011/09/machine-learning-text-feature-extraction-tf-idf-part-i/