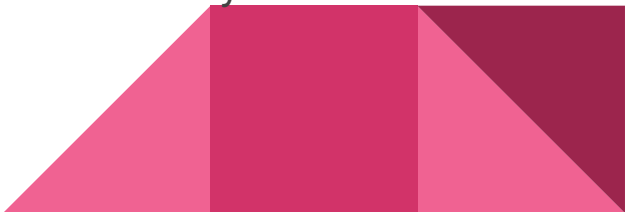


# Analyzing #POTUS Sentiment on Twitter to Predict Public Opinion on Presidential Issues

By: Jacob Handy  
Austin Karingada

# Project Description

- Goal: predict public opinion on a presidential policy by searching for sentiment patterns in past tweets using #POTUS.
  - Purpose: analyze twitter data with the Naïve Bayes model and search for patterns in keywords and the associated sentiment.
  - Related work: Predict popular trends in the #metoo movement & a study that identified sentiment using the presence of emojis
- 

# ML methods

- **Naive Bayes:**

- A naive Bayes classifier is an algorithm that uses Bayes' theorem to classify objects.
- Naive Bayes classifiers assume strong, or naive, independence between attributes of data points.

- **One-Hot Encoding:**

- A one hot encoding is a representation of categorical variables as binary vectors.

- **Support Vector Machines:**

- supervised learning models that analyze data used for classification and regression analysis.
- an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier

# Collecting the data

1. We used a Twython query with parameters:
  - a. Searching for #POTUS
  - b. Switch between mixed and recent results
  - c. 100 tweets at a time
  - d. Tweets in english
2. Preprocessed the tweets to be used nicer
3. One-hot encoded data into dictionary of id, classes, and sentiment
4. Wrote dictionary to csv file without label and id for calculating



# One hot encoding

- Presence of keyword marked as 1, while not present is 0
- Positive sentiment is 1, negative is 0

1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32					
2	0	1	0	1	0	0	1	0	0	1	1	1	0	0	0	0	1	0	0	1	1	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	0	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	1	0	1	1	1	1	1	0	0	0	1	0	1	1	1	1	0	1	0	0	1	1



# Organized data for Naive bayes algorithm

- Due to the nature of writing a dictionary to a csv file, the data was needed by column, not rows
- We solved this issues by creating a list of lists that formatted the data column-by-column to a row each
- This transformation can be exemplified as the data being converted from the one-hot encoding slide to last slide's table



# Naive Bayes Algorithm

- Simple and effective classification algorithm
- Supervised learning
- Popular uses include: spam filters, text analysis and medical diagnosis.
- Assumes that the probability of each attribute belonging to a given class value is independent of all other attributes
- Calculates the probability of each instance of each class and selects the highest probability





## Naive Bayes math

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp \left( -\frac{(x_i - \mu_y)^2}{2\sigma_y^2} \right)$$

# The process of Naive Bayes

## Before the prediction:

1. Preprocess the data into the table format from earlier
2. Split the data set with 67% for training set and 33% for test set
3. Separate data by classes to calculate the statistics for each class
4. Calculate the mean
5. Calculate the standard deviation
6. Collect the values

## Prediction:

1. Calculate probabilities using the equation on last slide
2. Summarize all the probabilities for each class
3. Make a prediction based on the best probability
4. Test the probabilities with the actual values
5. Get the accuracy as a percentage



67.55%

Accuracy

# Data Limitations

1. Twitter API plan:(free)
  - a. 100 tweets/request
  - b. 30 requests/min
  - c. 256 characters
  - d. Last 30 days
2. Very bad misspellings
  - a. Ex: Muler != Mueller
3. Lack of bigrams
4. Does not detect sarcasm



## Solved Limitations:

1. Variations: search first half of class word
2. Punctuation: replaced punctuation with nothing(Ex: ' ')
3. Capitalization: .lower() method

# Our Greatest Limitation

Sarcasm!

- Yes, sarcasm is a big problem in our algorithm.
  - However, there is not a good way of detecting sarcasm as people aren't even that good at it
  - One way is to identify positive and negative words in one string, but it is not very effective
  - Research is ongoing with CNNs
-

# Future Work

- Find most optimized algorithm between Naive Bayes and SVM
  - SVM proved faster in a similar sentiment analysis project
- Bernoulli Naive Bayes
  - Now that we dropped the neutral sentiment, the data is now binary
  - Improves accuracy in the assumption that data is binary
- Bigrams
  - Provides better context to the the sentiment



# SVMs

- Supervised Learning
- Can be used for both regression and classification but is used mainly for classification
- Objective is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. (N - the number of features)



# Pros and Cons of SVM

## Pros

- Accuracy

- Works well on smaller cleaner datasets

- It can be more efficient because it uses a subset of training points

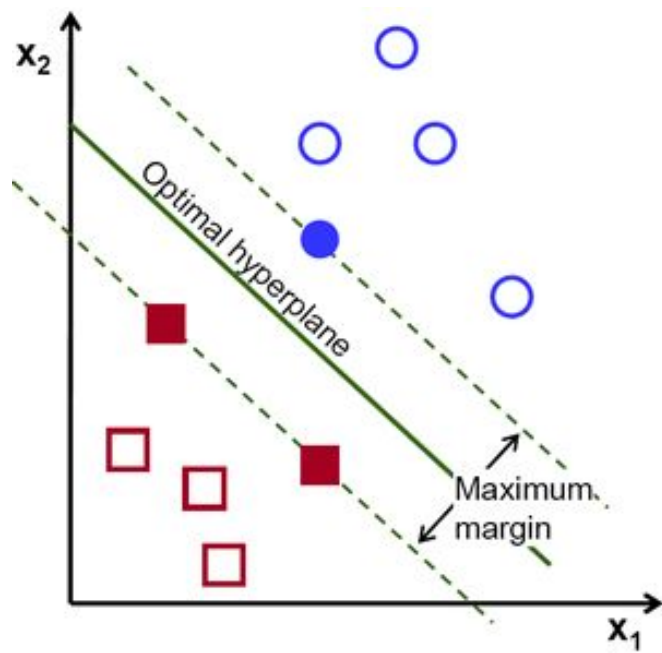
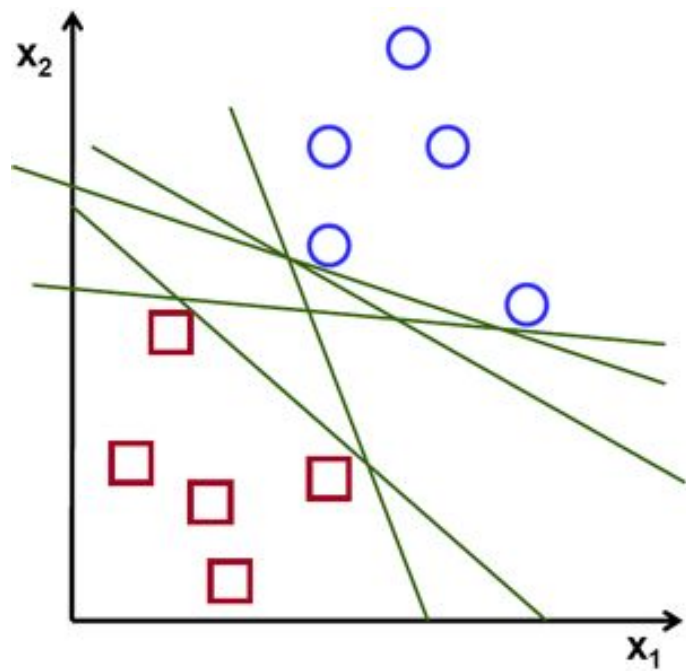
## Cons

- Isn't suited to larger datasets as the training time with SVMs can be high


- Less effective on noisier datasets with overlapping classes







# FAQ

1. How can we improve the accuracy?
    - a. Due to the new nature of our one-hot encoded data, a Bernoulli Naive Bayes implementation would work better
  2. Why weren't bigrams implemented?
    - a. We had problems implementing them as it complicated data formatting and we wanted to make sure we could implement individual keywords first
  3. Why didn't we upgrade our Twitter API plan?
    - a. Because its \$149 and we're broke
  4. How can we handle sarcasm?
    - a. Addressed in slide 14, but CNN models are pre-trained and used to extract sentiment, emotion, and personality features which captures the context of information
  5. What will be done between now and the report?
    - a. Finish implementing the SVM algorithm and compare time and accuracies
- 

# References

Twitter API.

Go, Alec, et al. "Twitter Sentiment Classification Using Distant Supervision." Cs.stanford.edu, Stanford University, cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf.

Berwick, R. "An Idiot's Guide to Support Vector Machines (SVMs)." Web.mit.edu, MIT, web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf.

Rish, I. "An Empirical Study of the Naive Bayes Classifier." Cc.gatech.edu, T.J. Watson Research Center, www.cc.gatech.edu/isbell/reading/papers/Rish.pdf.

Brownlee, Jason. "Naive Bayes Classifier From Scratch in Python." *Machine Learning Mastery*, 31 Aug. 2018, machinelearningmastery.com/naive-bayes-classifier-scratch-python/.

<https://github.com/ryanmcgrath/twython>





Questions?