



SENTIMENT ANALYSIS

PRESENTED BY :

ANKIT ARYA – 2017UCO1629

LAKSHAY DABAS – 2017UCO1624

HIMANSHU – 2017UCO1588

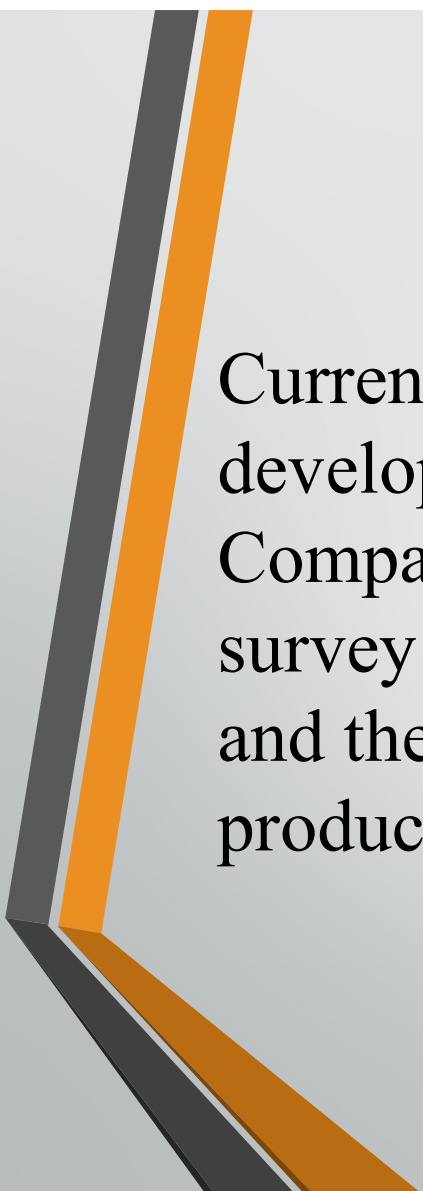
INTRODUCTION

Sentiment analysis is the automated process of analyzing text data and classifying opinions as *negative*, *positive* or *neutral*. Usually, besides identifying the opinion, these systems extract attributes of the expression e.g.:

Polarity: if the speaker express a *positive* or *negative* opinion,

Subject: the thing that is being talked about,

Opinion holder: the person, or entity that expresses the opinion.



Currently, sentiment analysis is a topic of great interest and development since it has many practical applications. Companies use sentiment analysis to automatically analyze survey responses, product reviews, social media comments, and the like to get valuable insights about their brands, product, and services.

Naive Bayes Classifier

The Naive Bayes Classifier is a well-known machine learning classifier with applications in Natural Language Processing (NLP) and other areas. Despite its simplicity, it is able to achieve above average performance in different tasks like sentiment analysis.

- Our goal is to correctly classify a review as positive or negative , these are two classes to which each document belongs.

$$\hat{c} = \operatorname{argmax}_{c \in C} P(c|d)$$

- In more mathematical terms, we want to find the most probable class given a document, which is exactly what the above formula conveys. **C** is the set of all possible classes, **c** one of these classes and d the document that we are currently classifying. We read **P(c|d)** as the probability of class **c**, given document **d**.

- The equation can be rewritten using the bayes rule , which is one of the most fundamental rules in machine learning.

$$\hat{c} = \operatorname{argmax}_{c \in \mathcal{C}} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

- **Note:** Since we want to maximize the equation we can drop the denominator, which doesn't depend on class c .

- Naive Bayes assumption: given a class c , the presence of an individual feature of our document is independent on the others.
- We consider each individual word of our document to be a feature. If we write this formally we obtain:

$$c_{NB} = \operatorname{argmax}_{c \in C} P(c) \prod_{f \in F} P(f|c)$$

- The Naive Bayes assumption lets us substitute $P(d|c)$ by the product of the probability of each feature conditioned on the class because it assumes their independence.

• Smoothing

- Imagine that you are trying to classify a review that contains the word 'stupendous' and that your classifier hasn't seen this word before.
- Naturally, the probability $P(w_i|c)$ will be 0, making the second term of our equation go to negative infinity!
- This is a common problem in NLP but thankfully it has an easy fix: smoothing. This technique consists in adding a constant to each count in the $P(w_i|c)$ formula, with the most basic type of smoothing being called add-one (Laplace) smoothing, where the constant is just 1.

$$\hat{P}(w_i|c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c) + 1)} = \frac{\text{count}(w_i, c) + 1}{(\sum_{w \in V} \text{count}(w, c)) + |V|}$$



Training and Testing

We will test our model on a dataset with 1000 positive and 1000 negative **movie reviews**. Each **document** is a review and consists of one or more sentences.

We split the data into a **training set** containing 90% of the reviews and a **test set** with the remaining 10%. As the name implies, the former is used for training the model with our train function, while the latter will give us an idea how well the model generalizes to unseen data.

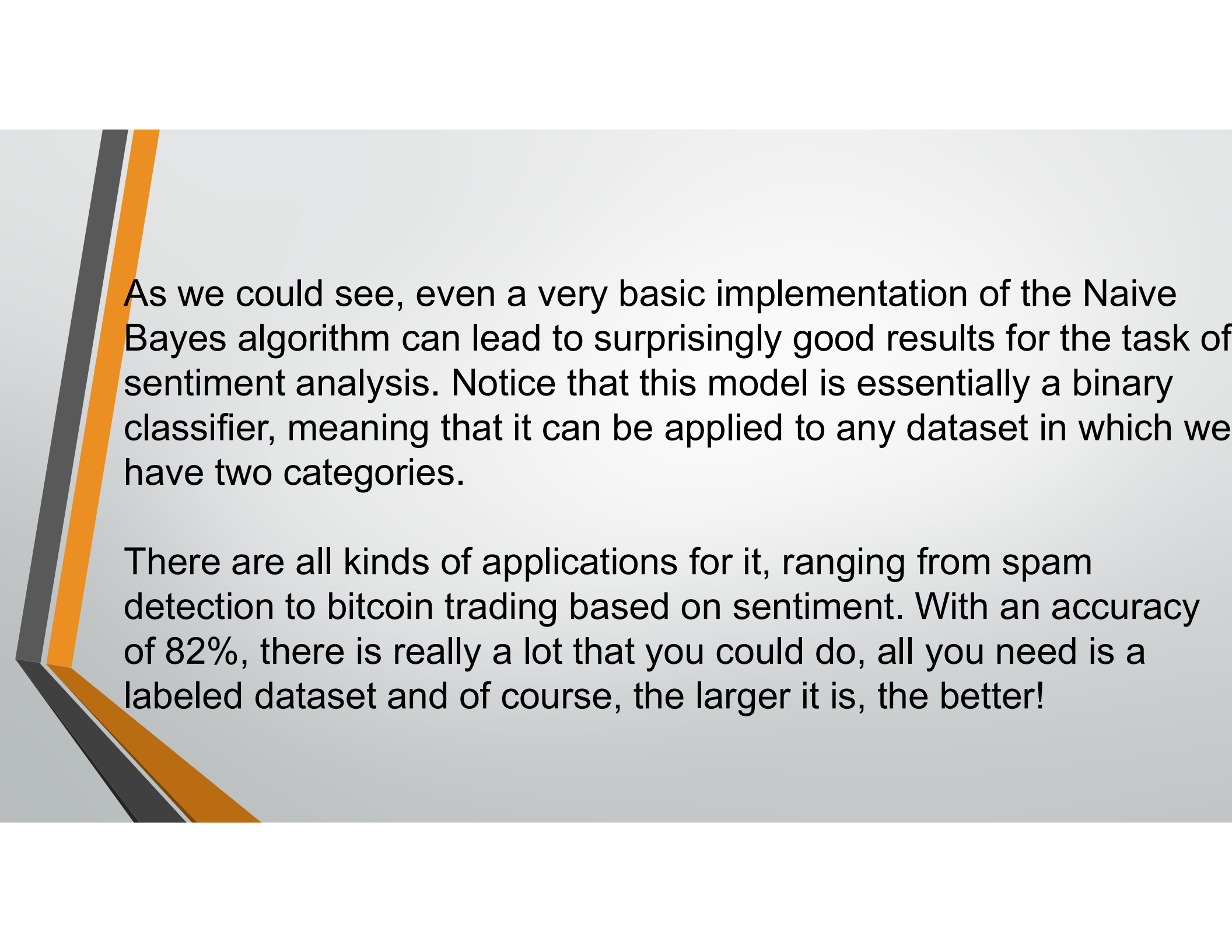
CONCLUSION

By using Naive Bayes classifier for doing sentiment analysis we were able to achieve an accuracy of 82.17822%

Our model correctly classify at least 4 out of 5 reviews, a very nice result.

We also see that training and predicting both together take around 1 second which is a relatively low runtime for a dataset with 2000 reviews.

```
Predicted correctly 166 out of 202 (82.17822%)  
Ran in 1.054 seconds
```



As we could see, even a very basic implementation of the Naive Bayes algorithm can lead to surprisingly good results for the task of sentiment analysis. Notice that this model is essentially a binary classifier, meaning that it can be applied to any dataset in which we have two categories.

There are all kinds of applications for it, ranging from spam detection to bitcoin trading based on sentiment. With an accuracy of 82%, there is really a lot that you could do, all you need is a labeled dataset and of course, the larger it is, the better!