Depression Detection in Social Media by Analyzing User’s Sentiment (Naive Bayes)

Group G

ANGGOTA KELOMPOK

DARREN VERNON RIOTA

00000032552

JERICHO CRISTOFEL SIAHAYA

00000032932

M. RIZKY AZZAKKY

00000033354

RICKY NG

00000032666

Latar Belakang Permasalahan

The **basis** of our research.

Depresi

- Sebuah **penyakit gangguan mental** yang dialami oleh **264 juta orang** worldwide
- Berbagai macam faktor seperti **faktor sosial, psikologis, biologis** dan sebagainya
- Membunuh sekitar **800,000 orang annually (suicide)**

Sentiment Analysis

- Penggunaan **sosial media** yang meningkat memudahkan **pengguna** untuk mengekspresikan **perasaannya**
- **Unggahan pengguna** mencerminkan perasaan atau emosi **(sentiment)**
- **Sentimen dianalisis** untuk **mengklasifikasi perasaan** pengguna

Objective

Membangun sebuah **NLP Tool** (bisa diimplementasikan dalam bentuk Web Application ataupun API) yang menggunakan machine learning untuk mengklasifikasi **kesehatan mental** seseorang berdasarkan **unggahan sosial media** (Twitter) ke dalam 2 kategori; **depressed** or **not depressed.**

Our approach

Naive Bayes (Algoritma)

Naive Bayes (Theorem)

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Merupakan teorema yang bekerja pada **conditional probability**.

Conditional Probability adalah suatu probabilitas atau kemungkinan bahwa sesuatu akan terjadi berdasarkan kejadian-kejadian yang sudah terjadi sebelumnya.

Keunggulan **Naive Bayes** terhadap klasifikasi teks :

- Algoritma yang paling populer untuk klasifikasi teks
- Lebih simpel dan mudah untuk diimplementasikan
- Memiliki performa yang cukup cepat
- Memiliki tingkat keberhasilan lebih tinggi daripada algoritma lainnya.

How it works?

Input dataset (sentence/documents)

"I have a dog" — consider **positive** 😊

"My brother hates my dog" — consider **negative** 😟

Term of frequency from both class

This is our model

Word	Positive	Negative
i	0.11	0.00
have	0.11	0.00
a	0.11	0.00
dog	0.11	0.11
my	0.00	0.22
brother	0.00	0.11
hates	0.00	0.11

How it works?

New input (sentence/documents)

"I love my dog"

— is this sentence **positive** 😊 or **negative** 😟

Using conditional probability aka Naive Bayes

- Probability of positive:

P(I love my dog | positive) = P(I | positive) x P(love | positive) x P(my | positive) x P(dog | positive)

- Probability of negative:

P(I love my dog | negative) = P(I | negative) x P(love | negative) x P(my | negative) x P(dog | negative)

How it works?

Remember our previous table?

Word	Positive	Negative
i	0.11	0.00
have	0.11	0.00
a	0.11	0.00
dog	0.11	0.11
my	0.00	0.22
brother	0.00	0.11
hates	0.00	0.11

Calculation from both probability

- Probability of positive:

P(I love my dog | positive) = 0.11 x **0** x **0** x 0.11

- Probability of negative:

P(I love my dog | negative) = **0** x **0** x 0.22 x 0.11

We can't have **zero** number, so we'll do the *smoothing.* 🥤

How it works?

Laplace aka Additive Smoothing

Formula:

$$\hat{\theta}_i = \frac{x_i + \alpha}{N + \alpha d} \qquad (i = 1, \ldots, d),$$

Output/result from smoothing

Word with zero number (or word from new sentence that doesn't occur in both class):

- **P(love | positive)** = 1.11
- **P(i | negative)** = 2.22
- **P(love | negative)** = 1.11
- **P(my | positive)** = 3.33

We can finally calculate our probability in peace. ✌️

How it works?

Calculation from both probability, after smoothing

- Probability of positive:

P(I love my dog | positive) = 0.11 x 1.11 x 3.33 x 0.11

= **0.04472523** 😊

- Probability of negative:

P(I love my dog | negative) = 2.22 x 1.11 x 0.22 x 0.11

= **0.05963364** 😟

Output

Since the result of **negative** probability is higher than positive probability, then we consider “I love my dog” is a negative sentence, based on bayes calculation.

“I love my dog” = 😟

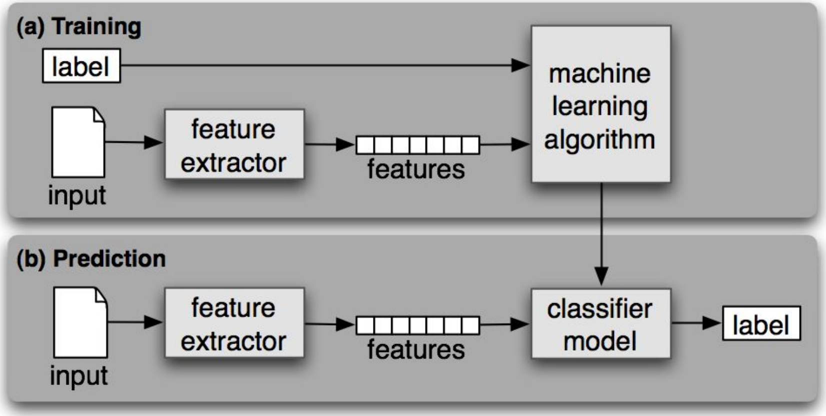
It’s weird, but that’s how Naive Bayes works. That’s why it’s called **“naive”** because it assumes that each input variable is **independent**. In other words, it doesn’t care about the context of a sentence but only the calculation through the data/term frequency.

Preparation

1. Dataset
   - **Sentiment140 from Kaggle**
   - 1,600,000 tweet with balanced negative and positive label/class
2. Programming Language
   - **Python 3**
   - Jupyter Notebook for live code, equations and visualizations.
3. Library
   - Numpy
   - Pandas
   - scikit-learn
   - Matplotlib, Seaborn
   - Flask (for deployment)

Implementation (with steps)

1. **Data preparation**
2. **Data cleansing**
3. **Splitting**
4. **Training (modelling) & Testing (validating)**
5. **Evaluation**



Data Preparation

Dataset Sentiment140 dari Kaggle

berisi 1.600.000 *tweets*

Fields : Target, ID, Date, Flag, User, Text

Yang akan digunakan : **Target** dan **Text**

https://www.kaggle.com/kazanova/sentiment140

Data Cleansing

Tujuannya : Membuat data yang berupa teks menjadi lebih dapat dipahami oleh komputer.

1. **Remove Punctuation**
2. **Remove Emoji**
3. **Remove Hyperlink**
4. **Convert into Lowercase**
5. **Tokenization**
6. **Remove Stopwords**

Data Cleansing

target	text
0	@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer. You shoulda got David Carr of Third Day to do it. ;D
0	is upset that he can't update his Facebook by texting it... and might cry as a result School today also. Blah!
0	@Kenichan I dived many times for the ball. Managed to save 50% The rest go out of bounds
0	my whole body feels itchy and like its on fire
0	@nationwideclass no, it's not behaving at all. i'm mad. why am i here? because I can't see you all over there.
0	@Kwesidei not the whole crew
0	Need a hug
0	@LOLTrish hey long time no see! Yes.. Rains a bit ,only a bit LOL , I'm fine thanks , how's you ?
0	@Tatiana_K nope they didn't have it
0	@twittera que me muera ?

↓

text	label
switchfoot awww thats bummer shoulda got david carr third day	0
upset cant update facebook texting might cry result school today also blah	0
kenichan dived many times ball managed save 50 rest go bounds	0
whole body feels itchy like fire	0
nationwideclass behaving im mad cant see	0
kwesidei whole crew	0
need hug	0
loltrish hey long time see yes rains bit bit lol im fine thanks hows	0
tatianak nope didnt	0
twittera que muera	0

Splitting

Untuk membuat model yang semakin **akurat**, dibutuhkan dataset yang lebih banyak pada fase training.

95%
Training

To train the model

5%
Testing

To determine the accuracy of the model

Training and Testing

Training (Modelling)

- Pipeline
  - Term Frequency (TF-IDF)
    - n_gram range = (1,3)
  - Naive Bayes (Multinomial)
    - Laplace/ Additive Smoothing with **α = 10**
- Elapsed time: **0.23 s**

***n-gram** is a contiguous sequence of n items from a given sample of text or speech.*

Testing (Validating)

- 5% dari keseluruhan dataset
- Accuracy score: 0.80 **(80%)**

Evaluation

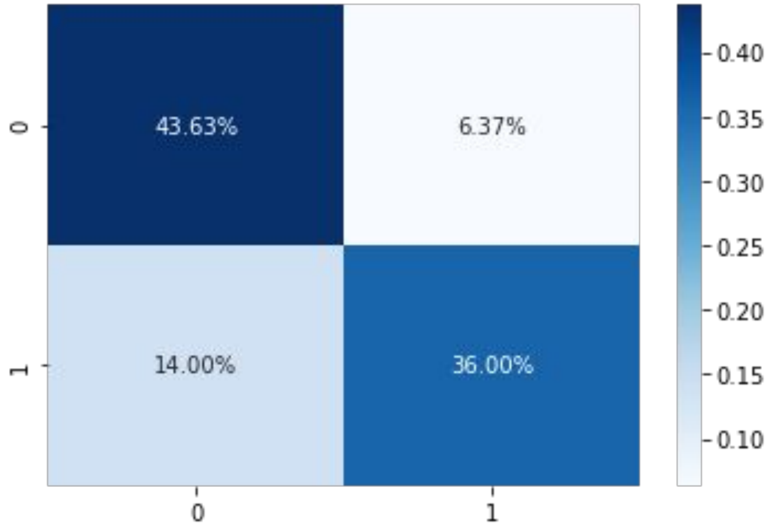
Recall pada class positive (yang tidak depresi) dipilih sebagai metrik acuan.

Classification Report:

	Precision	Recall	F1 Score
0	0.76	0.87	0.81
1	0.85	0.72	0.78
Accuracy	0.8		

Di sini, **False Negative** memiliki *cost* yang lebih tinggi dibanding False Positive. (FN > FP)

Confusion Matrix:



Let’s check out the

Demo

⬇️

ddnb.herokuapp.com

Kesimpulan

1. **Naive Bayes** merupakan algoritma klasifikasi yang paling populer
2. Naive Bayes menghitung probability tiap teks secara **independent**
3. **n_gram** digunakan untuk membuat training model semakin bervariatif
4. Untuk membuat model yang akurat, maka diperlukan **dataset training** yang lebih banyak
5. Pada studi kasus *depression detection* ini **False Negative** memiliki cost yang lebih tinggi ketimbang False Positive

Demo: **ddnb.herokuapp.com**

Live code: **gg.gg/ddnb-code**

Pembagian Tugas

Jericho Cristofel Siahaya

- Training (modelling) & Testing (validating)
- Evaluation
- Deployment (demo)

Darren Vernon Riota

- Riset mengenai latar belakang permasalahan
- Riset mengenai algoritma Naive Bayes

Muhammad Rizky Azzakky

- Evaluation
- Riset mengenai perbandingan Naive Bayes dengan algoritma lain

Ricky Ng

- Data preparation
- Data cleansing

Thank you.