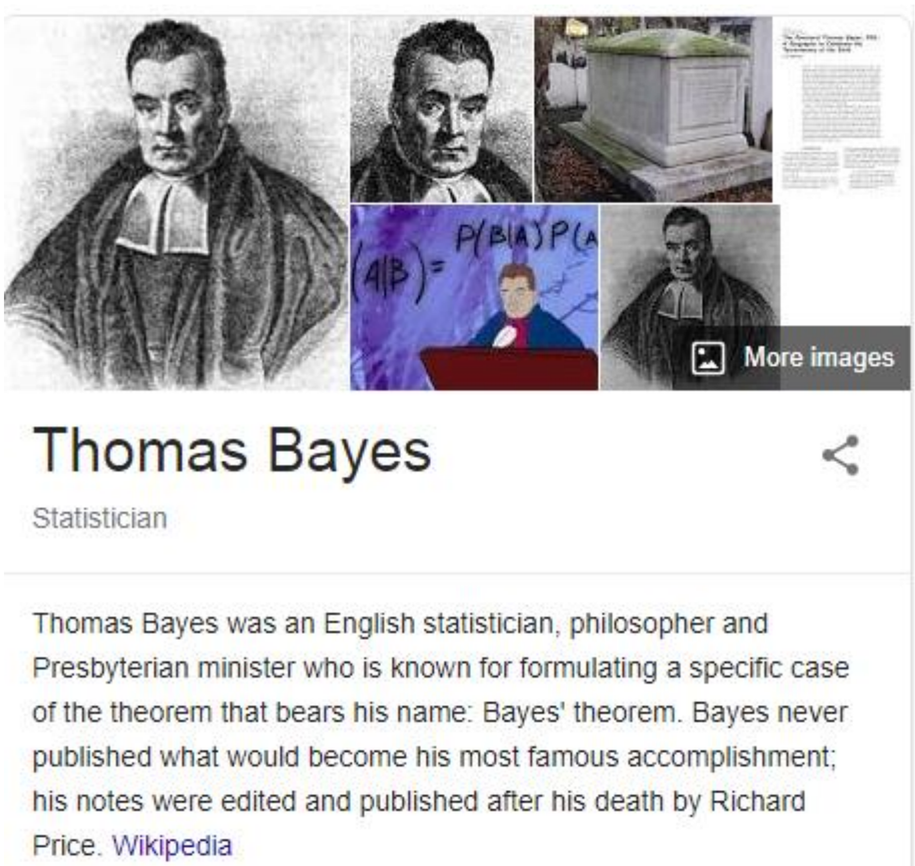
NAÏVE BAYES CLASSIFIER

Naive Bayes classifier is a machine learning method that utilizes probability and statistical calculations proposed by British scientist Thomas Bayes, which predicts future probabilities based on previous experience.



Example of Discrete Probability Distribution (Binomial)

Experiment :
The balanced coin is tossed four times.

Question :
Determine the probability distribution for the number of edges of the image that appear.

Solution :

- The sample space is S = {GGGG, GGGA, GGAG, GGAA, …, AAAA}.
- The number of sample points of S is n(S) = 2^4 = 16.
- From this experiment, several random variables can be defined which are able to map the sample space into real numbers or written as
  X : S → R.
- One of the random variables that can be made is
  X = the number of sides of the image that appear.
- Therefore, a numerical value of 0, 1, 2, 3 or 4 can be assigned to each sample point, namely:

$$X(GGGG) = 4,$$
$$X(GGGA) = 3, X(GGAG) = 3, X(GAGG) = 3,$$
$$X(GGAA) = 2, X(GAGA) = 2, X(GAAG) = 2,$$
$$X(GAAA) = 1$$

$$X(AGGG) = 3,$$
$$X(AGGA) = 2, X(AGAG) = 2, X(AAGG) = 2,$$
$$X(AGAA) = 1, X(AAGA) = 1, X(AAAG) = 1,$$
$$X(AAAA) = 0$$

- Thus, X = {0, 1, 2, 3, 4}

No.	I	II	III	IV	Experimental Results (= sample point)	X(S=s)	
1	G	G	G	G	GGGG	X(GGGG) =	4
2	G	G	G	A	GGGA	X(GGGA) =	3
3	G	G	A	G	GGAG	X(GGAG) =	3
4	G	A	G	G	GAGG	X(GAGG) =	3
5	G	G	A	A	GGAA	X(GGAA) =	2
6	G	A	G	A	GAGA	X(GAGA) =	2
7	G	A	A	G	GAAG	X(GAAG) =	2
8	G	A	A	A	GAAA	X(GAAA) =	1
9	A	G	G	G	AGGG	X(AGGG) =	3
10	A	G	G	A	AGGA	X(AGGA) =	2
11	A	G	A	G	AGAG	X(AGAG) =	2
12	A	A	G	G	AAGG	X(AAGG) =	2
13	A	G	A	A	AGAA	X(AGAA) =	1
14	A	A	G	A	AAGA	X(AAGA) =	1
15	A	A	A	G	AAAG	X(AAAG) =	1
16	A	A	A	A	AAAA	X(AAAA) =	0

- The probability distribution for the number of sides of the image that appears is:

x	0	1	2	3	4
f(x) = P(X=x)	1/16 = 0.06	4/16 = 0.25	6/16 = 0.375	4/16 = 0.25	1/16 = 0.06

Bayesian Probability

- In the previous section, we calculated probabilities with the frequency of occurrences that can be repeated.
- In the Bayesian view, we want to quantify the uncertainty for events that may be difficult to repeat.
- Suppose we want to know what is the probability that Mars is habitable.

- This is something that cannot be quantified with frequency, nor an event that can be repeated (going to Mars, how many people are alive).
- However, of course we have an initial assumption (prior).
- With a new powerful tool, we can collect new data about Mars.
- With these data, we correct our opinion about Mars (posterior).
- This causes a change in decision making.
- In this case, we want to be able to quantify the expression of uncertainty; and make revisions about uncertainties using new evidence.
- In Bayesian, probability values are used to represent the degree of confidence/uncertainty.

$$P(x \mid y) = \frac{P(y \mid x)P(x)}{P(y)} \tag{2.8}$$

- $P(x)$ is called prior, i.e. our initial knowledge/assumption.
- After we observe a new fact y (it can be a set of data or a data point/event), we change our assumptions.
- $P(y \mid x)$ is called the likelihood function.
- Likelihood function describes the probability of the data, for assumptions/knowledge about x that varies (x as a parameter that can be set).
- With the likelihood function, we correct our final opinion which can be used to make a decision (posterior).
- **In general Bayesian probability changes prior to posterior due to the new belief (likelihood).**

$$P(x \mid y) = \frac{P(y \mid x)P(x)}{P(y)} \tag{2.8}$$

- Intuitively, the posterior is affected by priors, meaning that it depends on the sample we have, because priors are obtained/inferred from the sample.
- This applies to machine learning, the quality of the resulting model depends on the quality of the training data.
- In general, we do not know all the information about a given situation and do not know all the probability information.
- For example, the probability $P(x \mid y)$ can be calculated by $P(x,y)$ / $P(y)$. However, we do not know how many events (x,y) occur at the same time.
- Therefore, we can use Bayesian theory to calculate probabilities with other information that we know.

Naïve Bayes

- Simple Naïve Bayesian Classifier is a simple probability classifier method based on the application of Bayes' Theorem with the assumption that the explanatory variables are independent.
- Naïve Bayes is a very simple supervised learning algorithm.
- The idea is **similar** to the Bayesian probability in Chapter 2.

id	outlook	temperature	humidity	windy	play (class)
1	sunny	hot	high	false	no
2	sunny	hot	high	true	no
3	overcast	hot	high	false	yes
4	rainy	mild	high	false	yes
5	rainy	cool	normal	false	yes
6	rainy	cool	normal	true	no
7	overcast	cool	normal	true	yes
8	sunny	mild	high	false	no
9	sunny	cool	normal	false	yes
10	rainy	mild	normal	false	yes
11	sunny	mild	normal	true	yes
12	overcast	mild	high	true	yes
13	overcast	hot	normal	false	yes
14	rainy	mild	high	true	no

Tabel 4.1. Contoh dataset *play tennis* (UCI machine learning repository)

	outlook		temperature			humidity			windy			play (class)	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	3	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								

Tabel 4.2. Frekuensi setiap nilai atribut

	outlook		temperature			humidity			windy			play (class)	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2/9	3/5	hot	2/9	3/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

Tabel 4.3. Probabilitas setiap nilai atribut

id	outlook	temperature	humidity	windy	play (class)
1	sunny	cool	high	true	no

Tabel 4.4. Contoh testing data *play tennis* [7]

$$\text{likelihood}(c_i) = P(c_i) \prod_{f=1}^{F} P(t_f|c_i) \tag{4.1}$$

- Formally, the Naïve Bayes equation for classification is given in equation 4.1 where c_i is a class value, C is a class (set), t is a feature (one feature, not a feature vector) and F is the number of features.
- We predict classes based on the probability of occurrence of feature values in that class.

$$\begin{aligned}
\text{likelihood}(play = yes) &= P(yes)P(sunny|yes)P(cool|yes)P(high|yes)P(true|yes) \\
&= \frac{9}{14} * \frac{2}{9} * \frac{3}{9} * \frac{3}{9} * \frac{3}{9} \\
&= 0.0053 \\
\text{likelihood}(play = no) &= P(no)P(sunny|no)P(cool|no)P(high|no)P(true|no) \\
&= \frac{5}{14} * \frac{3}{5} * \frac{1}{5} * \frac{4}{5} * \frac{3}{5} \\
&= 0.0206
\end{aligned}$$

$$P_{assignment}(c_i) = \frac{\text{likelihood}(c_i)}{\sum_{c_j \epsilon C} \text{likelihood}(c_j)} \tag{4.2}$$

$$\hat{c}_i = \underset{c_i \epsilon C}{\arg\max}\, P_{assignment}(c_i) \tag{4.3}$$

$$
\begin{aligned}
P_{assignment}(play = yes) &= \frac{\text{likelihood}(play = yes)}{\text{likelihood}(play = yes) + \text{likelihood}(play = no)} \\
&= \frac{0.0053}{0.0053 + 0.0206} \\
&= 0.205 \\
P_{assignment}(play = no) &= \frac{\text{likelihood}(play = no)}{\text{likelihood}(play = yes) + \text{likelihood}(play = no)} \\
&= \frac{0.0206}{0.0053 + 0.0206} \\
&= 0.795
\end{aligned}
$$

- Since $P_{assignment}(play = no) > P_{assignment}(play = yes)$, it is decided that the class for the *unseen example* is ***play = no***.
- The classification process for new data is **the same as** the classification process for testing data, i.e. we want to guess the data class.
- Because the model managed to guess the class on the training data correctly, the accuracy of the model is 100% (coincidentally, there is only one example).
- **A large difference in likelihood indicates confidence, while a small difference indicates uncertainty.**

Application Example

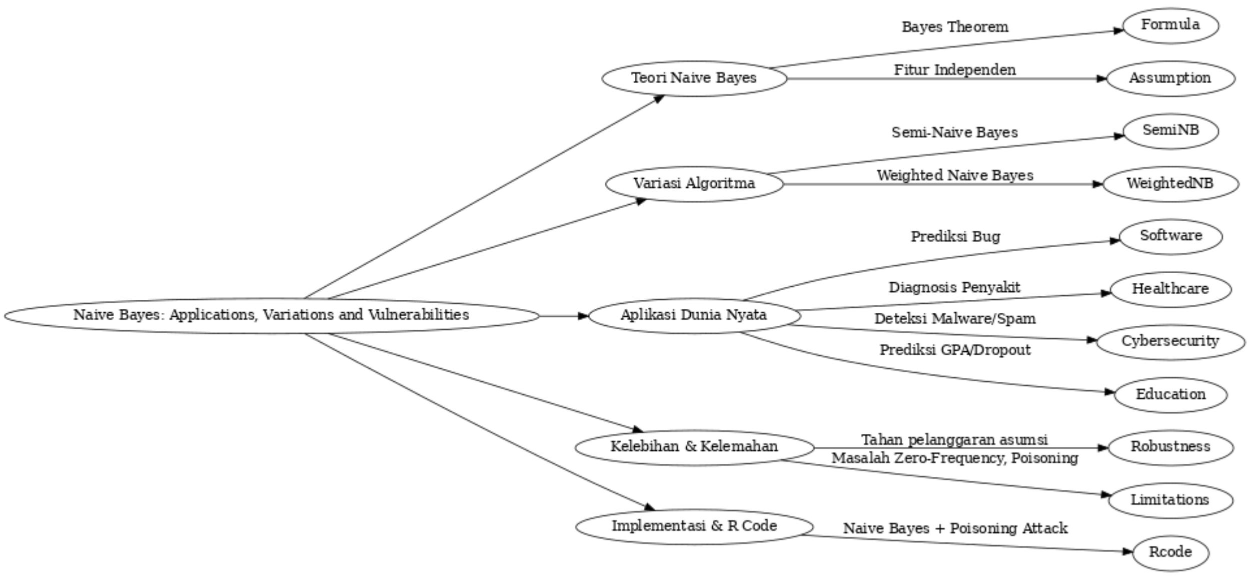
Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation

Naïve Bayes (NB) is a well-known probabilistic classification algorithm. It is a simple but efficient algorithm with a wide variety of real-world applications, ranging from product recommendations through medical diagnosis to controlling autonomous vehicles. Due to the failure of real data satisfying the assumptions of NB, there are available variations of NB to cater general data. With the unique applications for each variation of NB, they reach different levels of accuracy. This manuscript surveys the latest applications of NB and discusses its variations in different settings. Furthermore, recommendations are made regarding the applicability of NB while exploring the robustness of the algorithm. Finally, an attempt is given to discuss the pros and cons of NB algorithm and some vulnerabilities, with related computing code for implementation.

Keywords: Naïve Bayes, Probabilistic classification, Machine learning vulnerabilities, R code snippets

(Wickramasinghe, I., Kalutarage, H. Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. *Soft Comput* **25**, 2277–2293 (2021). https://doi.org/10.1007/s00500-020-05297-6**)**

Artikel ini merupakan kajian literatur yang mendalam tentang algoritma *Naive Bayes Classifier* (NBC), salah satu metode klasifikasi probabilistik yang paling sederhana namun efektif. Meskipun NBC bekerja berdasarkan asumsi fitur-fitur independen, kenyataannya, asumsi tersebut sering dilanggar dalam data dunia nyata. Oleh karena itu, penelitian ini membahas ragam aplikasi NBC, variasi-modifikasi algoritma (seperti *semi-naive* dan *weighted naive bayes*), analisis kekuatan dan kelemahan NBC, serta kode implementasi menggunakan bahasa R untuk eksperimen simulasi klasifikasi dan serangan *data poisoning*.



Pendahuluan

Naive Bayes berkembang seiring dengan kemajuan *low-cost computing* dan ledakan data. NBC menjadi favorit dalam klasifikasi data karena kecepatan tinggi, kesederhanaan implementasi, dan efisiensi baik pada dataset besar maupun kecil. Namun, asumsi utamanya — bahwa semua fitur bersifat *kondisional independen* — sering kali tidak realistis dalam kasus dunia nyata. Studi ini mengeksplorasi modifikasi terhadap NBC, kinerjanya dalam berbagai domain praktis, serta risiko-risiko yang muncul seperti kerentanan terhadap input berbahaya.

Metodologi

Penulis melakukan *systematic literature review* terhadap jurnal-jurnal terkini dan studi akademik yang relevan dengan penerapan dan pengembangan NBC. Fokus utama kajian meliputi:

- Struktur dasar teori NBC

- Modifikasi algoritmik:
  - *Semi-Naive Bayes* → mengurangi asumsi independensi
  - *Weighted Naive Bayes* → memberi bobot pada fitur penting
- Aplikasi NBC di berbagai sektor (software, kesehatan, keamanan siber, pendidikan)
- Eksperimen simulasi dengan R, termasuk:
  - Implementasi klasifikasi NBC
  - Simulasi serangan poisoning untuk melihat robustnes

Hasil dan Pembahasan

1. Variasi Algoritma

- **Semi-Naive Bayes**: Pendekatan ini memperhatikan ketergantungan antar fitur. Contoh teknik: AODE, TAN, dan KDB.
- **Weighted Naive Bayes**: Memberikan bobot berdasarkan kekuatan prediktif fitur. Teknik pembobotan meliputi *Gain Ratio*, *KL Divergence*, dan *MCMC*.

2. Aplikasi Dunia Nyata

- 🖥️ **Software Engineering**
  NBC digunakan secara luas untuk prediksi bug perangkat lunak. Studi NASA PROMISE menunjukkan NBC sebagai algoritma dominan dalam 47% penelitian serupa.

- 🛡️ **Healthcare**
  Digunakan untuk prediksi penyakit jantung, liver, dan kanker. Sering mengungguli metode lain dari segi kecepatan dan akurasi.

- 🔐 **Cybersecurity**
  Digunakan dalam spam filtering, malware detection, phishing attack detection, hingga keamanan IoT.

- 🎓 **Education**
  Untuk prediksi GPA (Grade Point Average) atau IPK, dropout

mahasiswa, dan pemantauan personalisasi pembelajaran. NBC menunjukkan performa lebih tinggi dibanding decision tree dan regresi.

3. Robustness dan Kelemahan

- **Kelebihan:**
  - Tahan terhadap *missing values*.
  - Cepat dilatih dan tidak mudah overfitting.
  - Tetap bekerja baik meskipun asumsi independensi dilanggar.
- **Kelemahan:**
  - Masalah *zero-frequency*: jika fitur belum pernah muncul dalam pelatihan.
  - Rentan terhadap *data poisoning attack* jika input tidak divalidasi.
  - Kurang optimal jika fitur sangat saling tergantung.

Kesimpulan

Naive Bayes tetap menjadi algoritma klasifikasi yang efisien dan kompetitif. Meskipun asumsi dasar sering dilanggar dalam konteks data nyata, performanya tetap baik berkat sifat dasar yang probabilistik dan kesederhanaannya. Modifikasi seperti *Weighted Naive Bayes* dan *Semi-Naive Bayes* terbukti meningkatkan akurasi. NBC juga menunjukkan ketahanan terhadap *missing data* dan cocok digunakan dalam berbagai sektor. Namun, penting untuk memahami kerentanannya terutama dalam penerapan pada sistem terbuka dan real-time, sehingga mitigasi risiko harus dipertimbangkan.