مجلة جامعة دمشق للعلوم الأساسية
مجلد 40- عدد 2 (2024): 50-66

Damascus University Journal for the Basic Sciences
Vol 40 No.1 (2024): 50 -66

مقارنة بين طريقة مصنف بايز بحالتيها البسيطة والمحسنة

نزار احمد التنجي * د. عزات عمر قاسم **

* طالب دراسات عليا – كلية العلوم – جامعة دمشق.
nizar.altounji@damascusuniversity.edu.sy.

** قسم الإحصاء الرياضي – كلية العلوم – جامعة دمشق.
Izzat.kassem@damascusuniversity.edu.sy.

الملخص

نظراً لكثرة التطبيقات التي تكون فيها المسألة الأساسية هي تصنيف مشاهدات جديدة في فئات معلومة، تم وضع العديد من طرائق التصنيف الموجه التي تحل هكذا مسائل بالاعتماد على مشاهدات مأخوذة مسبقاً تُستخدم لبناء دالة التصنيف، وأهم هذه الطرق هي طريقة مصنف بايز NB.

تطرق هذا البحث للتعاريف الأساسية المتعلقة بعملية التصنيف الموجه ولطريقة مصنف بايز حيث تم تعرف على آلية عملها وتعريف الحالة المحسنة لها والمسماة بمصنف بايز المرن FNB.

وتم إجراء مقارنة تطبيقية بين الحالة البسيطة والمحسنة بالاعتماد على عدة قواعد بيانات مختلفة الأحجام والأبعاد، تّبين من خلال النتائج تفوق طريقة مصنف بايز المرن على طريقة مصنف بايز في معظم التطبيقات، بالإضافة لذلك أظهرت النتائج أهمية التوزيع الاحتمالي للمتغيرات المستقلة وحجم عينة التدريب للوصول لدقة تصنيف أعلى ولتوضيح الفرق بين دقتي تصنيف الطريقتين وأيهما الأفضل للاستخدام.

الكلمات المفتاحية: نظرية التعلم الإحصائي – تصنيف موجه – تعلم الآلة – مصنف بايز – بايز المرنة – مقدر النواة.

تاريخ الإيداع: 2022/08/29
تاريخ الموافقة: 2022/11/20

ISSN (online): 2789-6366
http://journal.damascusuniversity.edu.sy/index.php/index/index

Comparison Between Classical and Advanced Naïve Bayes Classifier

Nizar Ahmad Altounji[1] **Dr. Izzat Omar Kassem[2]**

* Postgraduate Student - Faculty of Science - Damascus University.
nizar.altounji@damascusuniversity.edu.sy.

** Department of Mathematical Statistics - Faculty of Science -Damascus University.
Izzat.kassem@damascusuniversity.edu.sy.

Abstract:
Due to the quite number of applications which the main problem of it is to classify new observations into known groups, several supervised classification methods were set to solve such problems depending on a dataset used to build a classification function, one of the most important methods is Naïve Bayes Classifier NB.
This research has addressed the basic definitions related to supervised classification and naïve bayes classifier, which it defines how the classifier works with related ideas, furthermore, it introduces the advanced status of NB, which is called Flexible Bayes FNB.
A practical comparison has been made between the basic NB and advanced FNB status using datasets of different applications in terms of size and dimensions, the results showed that FNB perform better than NB in most applications, also, it reveals the importance of probability distributions of the independent variables and the size of training data to achieve a higher accuracy and to show the difference of accuracies between NB and FNB, and which one is preferred to use.

Received :2022/08/29
Accepted:2022/11/20

Keywords: Statistical Learning Theory - Supervised Classification - Machine Learning - Naïve Bayes - Flexible Bayes - Kernel Density Estimation.

1. مقدمة:

يراود الباحثون في الكثير من الحالات تطبيقات عملية هدفها وضع مشاهدات جديدة في فئات معرفة مسبقاً، كتصنيف زهور جدد في زمرة من زمر الزهور المعلومة، معرفة لغة النص في عدد كبير من النصوص الإلكترونية، أو تشخيص الحالة المرضية لمصاب جديد، إلخ.

تكون هذه التطبيقات ذات هدف التصنيفي، أي المسألة الرئيسية هي القيام بتصنيف مشاهدات جديدة مجهولة الفئة في فئات الظاهرة المدروسة، يمكن للإنسان بمفرده أن يقوم يدوياً بهذه العملية ولكن بالطبع قد يُخطأ وسوف يستهلك الكثير من الوقت، فيتم الاعتماد على الإحصاء والرياضيات والبرمجيات الحاسوبية للقيام بهذا الأمر بدقة أعلى ووقت أقل بكثير.

يتم أخذ عينة من الظاهرة المدروسة (بيانات) يُعتمد عليها من أجل القيام ببناء دالة رياضية (خوارزمية) تقوم بعملية التصنيف، تدعى هذه العملية بالتصنيف الموجه Supervised Classification ونسمي الدالة بدالة تصنيف موجه واختصارا دالة تصنيف Classification Function.

نظراً لكثرة استخدام وأهمية هذه العملية، يوجد العديد من دوال تصنيف التي قام الباحثون على مر الزمان ببنائها، وسنناقش في هذا البحث إحدى هذه الطرق والتي تسمى بدالة مصنف بايز Naïve Bayes Classifier NB، والتي تمتاز بشهرتها وكثرة استخدامها في الكثير في المسائل.

2. أهمية البحث:

تمكن أهمية البحث في إعطاء المفاهيم الأساسية المتعلقة بعملية التصنيف الموجه ومناقشة دالة مصنف بايز التي تعتمد في آليتها على التوزيع الاحتمالي للمتغيرات العشوائية (الميّزات Features) للظاهرة المدروسة، حيث سنتناول الحالة الكلاسيكية والحالة المحسنة للدالة التي تدعى بدالة بايز المرنة Flexible Bayes FNB، لدراسة فعالية الدالة بالقيام بعملية التصنيف.

بالإضافة لذلك ستتم المقارنة بين الحالتين لمعرفة أيهما الأفضل بالنسبة لدقة التصنيف.

3. مشكلة البحث:

بما أن معظم التطبيقات المدروسة يكون التوزيع الاحتمالي فيها غير معلوم، فإن دالة مصنف بايز في الحالة الكلاسيكية تفرض توزيع احتمالي ما على متغيرات الظاهرة للقيام بحساب احتمالات شرطية لكل فئة من الفئات من أجل تصنيف المشاهدات، فقد يتم فرض توزيع احتمالي غير ملائم للظاهرة المدروسة مما يؤدي للحصول على دقة تصنيف ضعيفة.

سندرس مدى فعالية فرض بعض التوزيعات الاحتمالية الشهيرة وسنناقش الحالة المحسنة التي تكمن آليتها في تقدير التوزيع الاحتمالي باستخدام مقدر النواة Kernel Density Estimation KDE.

4. أهداف البحث:

أهداف البحث هي:

1. التعرف على دالة مصنف بايز.
2. التعرف على دالة بايز المرنة.
3. المقارنة بين الحالتين.

لنقدم الآن التعاريف والنظريات المتعلقة بأهداف البحث:

a. تعريف التصنيف الموجه Supervised Classification:

يكمن الهدف الأساسي في العملية بأن يتم التنبؤ بالفئة (الصنف) التي تنتمي لها مفردة جديدة من المجتمع المدروس وذلك ببناء نموذج (خوارزمية) تستخدم عينة مسحوبة (البيانات)، يمكن التعامل مع حالتين من التصنيف هما التصنيف الثنائي Binary Classification والتصنيف المتعدد Multi-classification [1] [2].

أي التنبؤ بقيمة متغير تابع Y يمثل الصنف، بالاعتماد على شعاع المتغير المستقل X التي تمثل عناصره خواص المشاهدة، وذلك حسب دالة تصنيف نرمز لها بـ g، تسمى Y صنف X، ونشير هنا إلى أن X هو شعاع المتغيرات المستقلة له d بُعد (عدد الخواص أو الميّزات).

b. تعريف دالة الخسارة Loss function:

تُعرف دالة الخسارة بأنها ثمن الخطأ في التصنيف لدالة تصنيف ما g، سواء كانت الحالة هي حالة تصنيف ثنائي أي تكون $g: \mathbb{R}^d \to \{0,1\}$ أو كانت الحالة هي حالة تصنيف متعدد أي تكون $g: \mathbb{R}^d \to \{1,2,\dots,m\}$ (m هي عدد الأصناف)، ونرمز لها بـ $Loss$ بالشكل:

$$Loss\big(\mathrm{Y}, g(\mathrm{X})\big) \coloneqq \begin{cases} 1; g(\mathrm{X}) \neq \mathrm{Y} \\ 0; otherwise \end{cases} \dots (1.1)$$

من أجل أي متجه عشوائي (X, Y) وأي دالة تصنيف g، حيث: $X: (\Omega, \mathcal{X}, P) \to (\mathbb{R}^d, \mathcal{R}^d)$

و$Y: \Omega \to \{0,1\}$ في حالة التصنيف الثنائي أو $Y: \Omega \to \{1,2,\dots,m\}$ في حالة التصنيف المتعدد.

حيث تدل $g(\mathrm{X}) \neq \mathrm{Y}$ على تصنيف X بفئة غير فئتها الحقيقية الممثلة بقيمة Y، أي صُنفت بشكل خاطئ [2] [3].

c. تعريف المخاطرة المتوقعة Expected risk والمخاطرة التجريبية Empirical risk:

تُعرف المخاطرة المتوقعة بأنها التوقع الرياضي لثمن الخطأ في التصنيف من أجل الدالة g، نرمز لها بـ $R(g)$ وتعطى بالعلاقة التالية [2] [3]:

$$R(g) \coloneqq E[Loss\big((\mathrm{X}, \mathrm{Y}), g(\mathrm{X})\big)] \dots (1.2)$$

ويمكن تقدير هذه الدالة بالاعتماد على المشاهدات المسحوبة، ويسمى تقدير الدالة بالمخاطرة التجريبية [2] [3]:

$$R_{emp}(g) \coloneqq \frac{1}{n}\sum_{i=1}^{n} Loss\big((x_{\mathrm{i}}, y_{\mathrm{i}}), g(x_{\mathrm{i}})\big) \dots (1.3)$$

d. دقة التصنيف Accuracy:

نعرف دقة التصنيف بأنها مقدار التصنيف الصحيح، أي تعطى بالعلاقة [4] [6]:

$$\mathrm{Accuracy}(g) \coloneqq 1 - R_{emp}(g) \dots (1.4)$$

e. مصنف بايز Naïve Bayes Classifier:

ليكن (X, Y) متجها عشوائياً يأخذ قيمه في $\mathbb{R}^{\mathrm{d}} \times \{1,2, \dots, m\}$، وليكن لدينا $\mathrm{X} = x$ مشاهدة جديدة مراد تصنيفها في إحدى الفئات، نصنف المشاهدة الجديدة في الفئة ذات الاحتمال الشرطي الأكبر، أي أولا نحسب [7] [9] [11] [12]:

$$P(\mathrm{Y} = j|\mathrm{X} = x) \propto P(\mathrm{Y} = j).\prod_{i=1}^{m} P(\mathrm{X}_i = x_i|\mathrm{Y} = j) \quad \forall \quad j = 1,2,\dots,m \dots (1.5)$$

ومنه يكون:

$$g_n(x) = arg\max_{1 \le j \le m} (P(\mathrm{Y} = j|\mathrm{X} = x)) \dots (1.6)$$

f. التوزيعات الاحتمالية ودالة تصنيف بايز:

نلاحظ أن دالة تصنيف بايز تعتمد على الكثافات الاحتمالية الشرطية للمتغيرات المستقلة حسب الفئات الموضوعة للقيام بالتصنيف، في معظم التطبيقات، لا يوجد لدينا معلومات قبلية عن التوزيع الاحتمالي لمتغيرات المسألة، لذلك يتم فرض توزيع احتمالي ما، يوجد ثلاث توزيعات احتمالية يتم الاعتماد عليها في أكثر التطبيقات، وهي التوزيع الطبيعي وتوزيع برنولي والتوزيع المتعدد، نستخدم عادة التوزيع الطبيعي مع المتغيرات المستمرة

وتوزيع برنولي والتوزيع المتعدد مع المتغيرات المنفصلة، نعتمد على المشاهدات المسحوبة لتقدير وسطاء التوزيع المفروض، ومنه نحسب الاحتمالات المطلوبة ونستخدمها في دالة التصنيف.

لنذكر التوزيعات الثلاث، علما أننا سنضعها بالحالة البسيطة (لمتغير عشوائي واحد فقط، لأننا سنستخدم فرض الاستقلال) [7] [9] [11] [12]:

1. Gaussian Naïve Bayes: [7]

تستخدم هذه الحالة مع المتغير العشوائي المستمر، وتكون دالة الكثافة الاحتمالية كما نعرفها بالشكل:

$$f_j(x) = \frac{1}{\sigma_j\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu_j}{\sigma_j}\right)^2} ; j = 1, \dots, m \dots (1.7)$$

حيث أن μ_j و σ_j هما المتوسط والانحراف المعياري للمتحول العشوائي X للفئة j.

2. Bernoulli Naïve Bayes: [7]

تستخدم هذه الحالة مع المتغير العشوائي المنفصل التي يأخذ قيمتين فقط، ويكون القانون الاحتمالي:

$$P(\mathrm{X} = x|\mathrm{Y} = j) = {p_j}^x(1 - p_j)^{1-x}; j = 1, \dots, m\ , x = 0,1 \dots (1.8)$$

تمثل p_j احتمال أخذ المتغير X القيمة واحد للفئة j.

3. Multinomial Naïve Bayes: [7]

تستخدم هذه الحالة مع المتغير العشوائي المنفصل التي يمثل عدد مرات وقوع كلاً من الأحداث الممكنة $\{1,2, \dots, k\}$، يكون القانون الاحتمالي بالشكل:

$$P(\mathrm{X} = x|\mathrm{Y} = j) = \frac{n!}{\prod_{i=1}^{k} x_i!} \prod_{i=1}^{k} {p_{ij}}^{x_i} ; j = 1, \dots, m\ , i = 1, \dots, k \dots (1.9)$$

تمثل p_{ij} احتمال وقوع القيمة i للمتغير X للفئة j.

g. طريقة بايز المرنة Flexible Bayes:

لا تختلف طريقة بايز المرنة FNB عن طريقة مصنف بايز NB بالمبدأ، إنما تعتمد على تقدير دالة الكثافة الاحتمالية لكل متغير عشوائي $\mathrm{X}_1, \dots, \mathrm{X}_d$ باستخدام مقدر النواة KDE بهدف تحسين عملية التصنيف من خلال الاعتماد على تقدير التوزيع الحقيقي بدلا من فرض توزيع قد يكون لا يلائم المتغير العشوائي كفرض التوزيع الطبيعي مثلاً، يجدر الذكر أن في حال كان التوزيع الاحتمالي لمتغير عشوائي ما معلوم فبالطبع يكون استخدام هذا التوزيع فرضاً يؤدي إلى تصنيف أفضل من التقدير، لكن كما قلنا سابقا أن معظم الحالات التطبيقية يكون التوزيع مجهولاً [9] [12].

h. تعريف مقدر النواة Kernel density estimation:

طريقة النواة هي طريقة تقدير إحصائية لا معلمية لدالة الكثافة الاحتمالية لمتحول عشوائي كثافته الاحتمالية $f(x)$ باستخدام عينة عشوائية $\mathrm{X}_1, \dots, \mathrm{X}_n$ تأخذ قيمها في $\mathbb{R}^d$، تعطى بالعلاقة [5] [8] [10] [13]:

$$\hat{f}(x) = \frac{1}{n|h|} \sum_{i=1}^{n} K\left(\frac{x - \mathrm{X}_i}{h}\right) \dots (1.10)$$

حيث:

h مصفوفة عرض النافذة وتدعى Bandwidth Matrix، و K دالة النواة.

من أجل حالتنا التي سنتعامل فيها مع شعاع المتغيرات العشوائية المستقلة X بأخذ كل متغير عشوائي منه على حدى (وذلك لفرضية الاستقلال) للقيام بتقدير كثافته الاحتمالية، سوف نأخذ مقدر النواة بحالته البسيطة التي تكون فيها h عبارة عن قيمة واحدة تسمى عرض النافذة Bandwidth [8] [10] [13].

في الدراسات النظرية حول جودة تقدير $f(x)$ بـ $\hat{f}(x)$، نفترض أن $h = h(n)$ وأن $h \longrightarrow 0$ و$n.h \longrightarrow \infty$ عندما $n \longrightarrow \infty$ وأن K هي دالة كثافة احتمالية متناظرة.

يوجد العديد من أنواع دوال النواة ولكل منها خواص متعلقة بها، من دوال النواة المستخدمة عندما $d = 1$ [5] [8] [10] [13]:

1. دالة النواة الطبيعية: $K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \dots (1.11)$
2. دالة النواة المنتظمة: $K(x) = \frac{1}{2} 1_{\{-1 \le x \le 1\}} \dots (1.12)$
3. دالة النواة Epanechnikov: $K(x) = \frac{3}{4} \max\{1 - x^2, 0\} \dots (1.13)$

حيث أن 1_A في دالة النواة المنتظمة هي دالة الواحدة.

لا تختلف النتائج بشكل كبير باختلاف دالة النواة المستخدمة، بل يتعلق الأمر باختيار عرض النافذة h، إذ أنه الأساس في تغيير شكل دالة الكثافة المقدرة، وتلعب الدور نفسه الذي يلعبه طول الفئة في المضلع التكراري لذلك يكون الشرطان $h \longrightarrow 0$ و$n.h \longrightarrow \infty$ أساسيان لتفادي وقوع معظم المشاهدات في فئة واحدة ولضبط التباين والانحياز [5] [8] [9].

i. نظرية الاتساق القوي لدالة الكثافة الشرطية:

ليكن لدينا $P(\mathrm{Y}|\mathrm{X})$ دالة الكثافة الاحتمالية الشرطية لـ Y من أجل فضاء العينة $(\mathrm{X}, \mathrm{Y}) \in \mathbb{R}^{\mathrm{d}} \times \{1,2, \dots, m\}$، نقول عن المقدر $P(y|x)$ المبني باستخدام طريقة مقدر النواة أنه ذو اتساق قوي لـ$P(\mathrm{Y}|\mathrm{X})$ [9]:

$$P(y|x) \xrightarrow{a.s} P(\mathrm{Y}|\mathrm{X}) \dots (1.14)$$

من أجل:

1. $h(n) \longrightarrow 0$ من أجل $n \longrightarrow \infty$.
2. $n.h(n) \longrightarrow \infty$ من أجل $n \longrightarrow \infty$.

أخيراً، يجدر التنويه بأن في طريقة مصنف بايز، نكتفي بحساب وسطاء التوزيع المفروض لحساب الاحتمالات الشرطية من أجل كل مشاهدة جديدة، بينما في طريقة بايز المرنة تُعاد الحسابات من أجل كل مشاهدة، أي سيكون هناك تكلفة حسابية وزمنية، فإذا كان توزيع المتغير X معلوم، يُفضل استخدام هذا التوزيع فرضا لتفادي هكذا تكلفة [9].

5. أدوات البحث ومواده:

تم استخدام دالة مصنف بايز وبايز المرنة في عدة تطبيقات عملية باستخدام لغة البرمجة الإحصائية R بهدف التعرف على جودة تصنيف الدالة بالحالة الكلاسيكية (مصنف بايز) والمحسنة (مصنف بايز المرنة) والمقارنة بين الحالتين، حيث قُسمت بيانات كل تطبيق عملي عشوائيا لقسمين، الأول يستخدم لبناء الدالة والثاني لاختبار دقة التصنيف باستخدامها، وتم تكرار بناء كل دالة 20 مرة وأخذ المتوسط الحسابي والانحراف المعياري لدقات التصنيف الناتجة للمقارنة بشكل أفضل، حيث تم ذلك عن طريق بناء خوارزمية برمجية تقوم بالتكرار المطلوب، ثم استخدمنا اختبار T-test لاختبار فرضية تساوي متوسطي دقات التصنيف للحالة الكلاسيكية والحالة المحسنة لكل تطبيق عند مستوى معنوية 0.05، حيث تم استخدام دالة النواة الطبيعية وبعرض نافذة يساوي $h = \frac{1}{\sqrt{n}}$ بالنسبة للحالة المحسنة من الدالة (حيث أن n هو حجم العينة).

اختلفت طبيعة البيانات من حيث حجمها وعدد الميّزات فيها وعدد الأصناف، كان منها ظواهر من الواقع ومنها مولدة عشوائياً، وكانت جميع الميّزات فيها عبارة عن متغيرات عشوائية مستمرة، إذ أن جوهر الاختلاف بين الحالتين يكمن في هذا النوع من المتغيرات وليس في المتغيرات الفئوية.

استُخدمت الحزم البرمجية class، caret، catools، e1071، naivebayes للاستفادة من الدوال البرمجية الجاهزة فيها، وسُميت الدوال المبرمجة من قبل الباحث بـ Loop20_NB_FNB وnormalize، ويمكن الاطلاع على التعليمات البرمجية في الملحق.

تمت معالجة البيانات الواقعية عن طريق حذف القيم المفقودة ومعايرة المتغيرات المستقلة المستمرة بحيث أن يتغير مجال القيم دون التأثير على قرار التصنيف باستخدام المعايرة المسماة Standardization أو Normalization.

كانت ست تطبيقات من الواقع وثلاث مولدة عشوائياً من قبل الباحث، حيث تم توليد هذه البيانات بأربعة متغيرات مستمرة مولدة وفقا للتوزيع الطبيعي، وكان متغير الصنف يأخذ قيمتين (تصنيف ثنائي)، أول قاعدة بيانات تحوي 200 مشاهدة، الثانية تحوي 500 مشاهدة والأخيرة تحوي 1000 مشاهدة. يوضح الجدول التالي التطبيقات العملية المستخدمة:

الجدول (1): جدول توضيحي لقواعد البيانات المستخدمة:

اسم قاعدة البيانات	حجم العينة	عدد الميّزات	عدد الأصناف	مجال التطبيق	طبيعة البيانات
Breast Cancer Coimbra	116	10	2	الطب	ظاهرة واقعية
Dry Bean	13611	17	7	علم الحياة	ظاهرة واقعية
Ionosphere	351	34	2	علم الفضاء	ظاهرة واقعية
Magic Gamma Telescope	19020	10	2	علم الفضاء	ظاهرة واقعية
Maternal Health Risk	1014	6	3	الطب	ظاهرة واقعية
Raisin	900	7	2	علم الحياة	ظاهرة واقعية
بيانات الباحث – 1	200	4	2	_____	توليد عشوائي
بيانات الباحث – 2	500	4	2	_____	توليد عشوائي
بيانات الباحث – 3	1000	4	2	_____	توليد عشوائي

المصدر: تم إعداده من قِبل الباحث

6. النتائج والاستنتاجات:

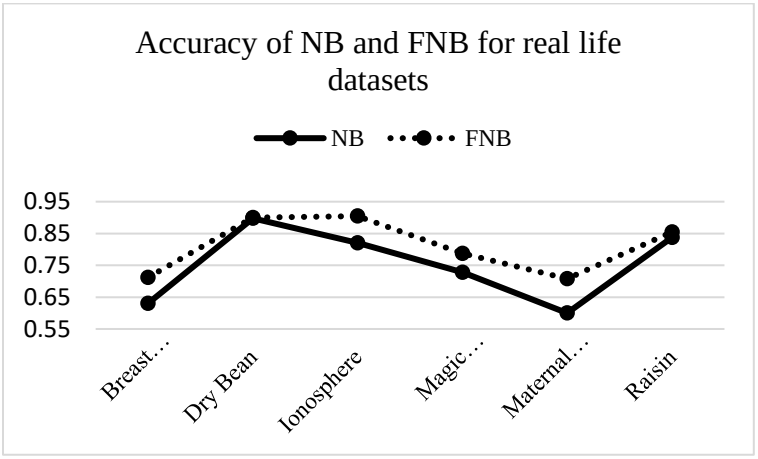
يعطي الجدول الآتي متوسط دقات التصنيف والانحراف المعياري لها (الانحراف المعياري $\pm$ المتوسط) الناتجة من بناء الدوال، بالإضافة لنتيجة اختبار T-test في العمود المسمى "رفض فرضية تساوي المتوسطين"، وتم كاتبة الحالة الأفضل من الدالتين لكل تطبيق بالخط الغامق:

الجدول (2): جدول نتائج بناء دالتي NB وFNB على قواعد البيانات التسع:

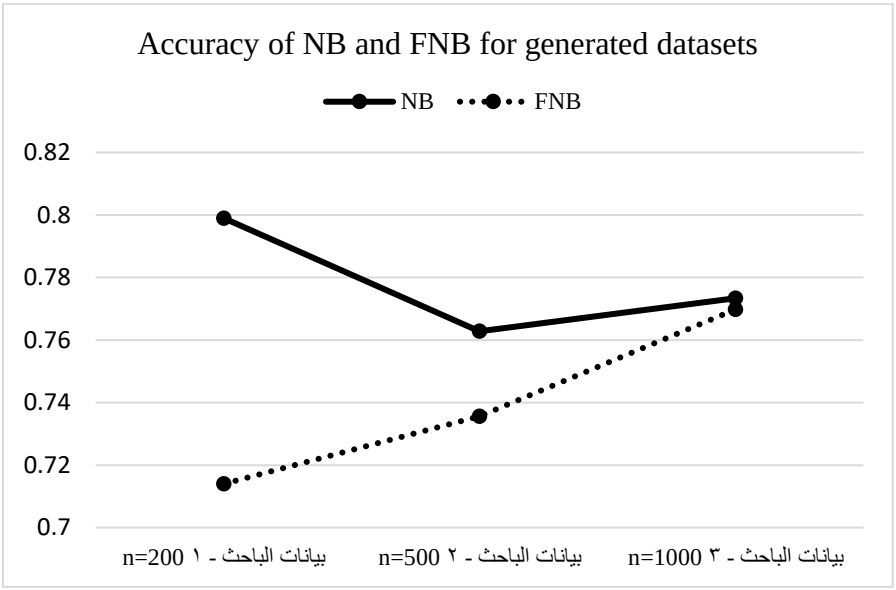
اسم قاعدة البيانات	NB	FNB	رفض فرضية تساوي المتوسطين
Breast Cancer Coimbra	0.631 ± 0.0717	$\mathbf{0.7121 \pm 0.0897}$	✓
Dry Bean	0.8976 ± 0.0043	$\mathbf{0.8998 \pm 0.0042}$	×
Ionosphere	0.821 ± 0.0344	$\mathbf{0.9045 \pm 0.0262}$	✓
Magic Gamma Telescope	0.7283 ± 0.0063	$\mathbf{0.787 \pm 0.0061}$	✓
Maternal Health Risk	0.6 ± 0.0239	$\mathbf{0.7079 \pm 0.0249}$	✓
Raisin	0.8373 ± 0.0174	$\mathbf{0.8544 \pm 0.0227}$	✓
بيانات الباحث – 1	$\mathbf{0.799 \pm 0.0586}$	0.714 ± 0.0592	✓
بيانات الباحث – 2	$\mathbf{0.7628 \pm 0.039}$	0.7356 ± 0.0261	✓
بيانات الباحث – 3	$\mathbf{0.7734 \pm 0.0236}$	0.7698 ± 0.0317	×

المصدر: تم إعداده من قِبل الباحث

ويوضح المخططين التاليين هذه النتائج:



الشكل (1): مخطط دقات التصنيف الناتجة لـ NB وFNB على قواعد البيانات الواقعية.



الشكل (2): مخطط دقات التصنيف الناتجة لـ NB وFNB على قواعد البيانات المولدة عشوائياً.

أعطت طريقة FNB دقة تصنيف أعلى من طريقة NB في التطبيقات الست الأولى وكانت نتيجة اختبار T-test أن نرفض فرضية تساوي متوسطي الطريقتين وقول إن أحدهما يملك دقة تصنيف أعلى وهي طريقة FNB، عدا التطبيق الثاني الذي دل فيه الاختبار على قبول فرضية التساوي، نلاحظ أن هذه التطبيقات هي مختلفة الحجم وعدد الميزات والأصناف، ومعظم الميّزات فيها لا يخضع لأحد التوزيعات الشهيرة المذكورة سابقاً (للتوزيع الطبيعي في الحالة العامة)، فكان استخدام مقدر النواة هو الأنسب وهو ما أدى إلى تفوق طريقة FNB.

وأعطت طريقة NB دقة تصنيف أعلى من طريقة FNB في التطبيقات الثلاث الأخيرة والتي هي مولدة عشوائياً، كان أول تطبيقين منهما له نتيجة اختبار T-test تدل على عدم تساوي المتوسطين، تدل هذه النتيجة على أن استخدام طريقة NB التي تفرض التوزيع الاحتمالي أفضل من طريقة FNB عند معرفة التوزيع الاحتمالي للمتغيرات والذي كان التوزيع الطبيعي لكل منها، ولكن كلما كبر حجم قاعدة البيانات قلّ الفرق بين دقتي تصنيف الطريقتين (تقارب دقة تصنيف مصنف بايز المرنة من دقة تصنيف مصنف بايز) وهو ما يتطابق مع نظرية الاتساق القوي لدالة الكثافة الشرطية.

7. التوصيات:

1. نوصي باستخدام طريقة FNB التي تعطي نتائج أفضل في معظم تطبيقات العملية، إذ أن معظم التطبيقات العلمية تشابه التطبيقات الستة الأولى في طبيعتها.
2. في حال وجود معلومات عن التوزيع الاحتمالي للمتغيرات المستقلة، يُفضل استخدام طريقة NB بالاعتماد على التوزيعات الاحتمالية المعلومة.
3. نوصي بإجراء أبحاث مماثلة تشمل استخدام دوال نواة مختلفة بنوافذ أُخرى لطريقة FNB.

8. الملحق:

a. التعليمات البرمجية المستخدمة في لغة R:

```
#Libraries:
library(ggplot2)
library(class)
library(caret)
library(e1071)
library(caTools)
library(naivebayes)
#Functions:
#1-Transform contin. vars. to a [0,1] range:
normalize <- function(x) {
return ((x - min(x)) / (max(x) - min(x))) }
#20 Implementations:
Loop20_NB_FNB <- function(fulldt) {
fulldt_Compare <- list(NULL)
for (i in 1:20) {
#Data splitting:
samp <- sample(1:nrow(fulldt),size=nrow(fulldt)*0.75,replace = FALSE)
train.fulldt <- fulldt[samp,-length(fulldt)]
test.fulldt <- fulldt[-samp,-length(fulldt)]
train.fulldt.label <- fulldt[samp,length(fulldt)]
test.fulldt.label <- fulldt[-samp,length(fulldt)]
#NB model:
starttime <- Sys.time();nb <- naiveBayes(x = train.fulldt,
y = train.fulldt.label);endtime <- Sys.time()
pred_nb <- predict(nb, newdata = test.fulldt)
confusionMatrix(table(pred_nb, test.fulldt.label))
acc_nb <- confusionMatrix(table(pred_nb, test.fulldt.label))$overall[1]
runtime_nb <- round(endtime - starttime,4)
#FNB model:
starttime <- Sys.time();fnb <- nonparametric_naive_bayes(x = as.matrix(train.fulldt,
nrow = dim(train.fulldt)[1], ncol = dim(train.fulldt)[2]),
y = train.fulldt.label,kernel = "gaussian",
bw = 1/sqrt(length(train.fulldt.label)));endtime <- Sys.time()
pred_fnb <- predict(fnb, newdata = as.matrix(test.fulldt,
nrow = dim(test.fulldt)[1], ncol = dim(test.fulldt)[2]))
confusionMatrix(table(pred_fnb, test.fulldt.label))
acc_fnb <- confusionMatrix(table(pred_fnb, test.fulldt.label))$overall[1]
```

```
runtime_fnb <- round(endtime - starttime,4)
#Models:
fulldt_Compare[[i]] <- data.frame("Method" = c("NB", "FNB"),
"Accuracy" = round(c(acc_nb, acc_fnb),4),
"RunTime" = c(runtime_nb, runtime_fnb))}
  #Mean/SD of 10 Implementations:
  fulldt_Compare_Total <- data.frame("Method" = c("NB", "FNB"),
"Mean_Accuracy" = rep(0, 2), "Sd_Accuracy" = rep(0, 2),
"Mean_RunTime" = rep(0, 2), "Sd_RunTime" = rep(0, 2))
allacc <- matrix(rep(0,40), nrow = 20, ncol = 2)
allruntime <- matrix(rep(0,40), nrow = 20, ncol = 2)
colnames(allacc) <- c("NB","FNB")
colnames(allruntime) <- c("NB","FNB")
for (j in 1:2) {
for (i in 1:20) {
allacc[i,j] <- fulldt_Compare[[i]][j,2]
allruntime[i,j] <- fulldt_Compare[[i]][j,3]}
fulldt_Compare_Total[j,2] <- round(mean(allacc[,j]),4)
fulldt_Compare_Total[j,3] <- round(sd(allacc[,j]),4)
fulldt_Compare_Total[j,4] <- round(mean(allruntime[,j]),4)
fulldt_Compare_Total[j,5] <- round(sd(allruntime[,j]),4)}
 hypo <- t.test(allacc[,2],allacc[,1])
return(list("Stats" = fulldt_Compare_Total, "T-test" = hypo, "Models" = fulldt_Compare))}
#BCC:
BCCdata <- read.csv(choose.files())
BCCdata$Class_name <- factor(BCCdata$Class_name)
BCCdata <- BCCdata[,-10]
BCCdata <- na.omit(BCCdata)
BCCdata[,1:9] <- as.data.frame(lapply(BCCdata[,1:9], normalize))
str(BCCdata)
result.BCCdata <- Loop20_NB_FNB(BCCdata)
#DryBean:
DryBeandata <- read.csv(choose.files())
DryBeandata$Class <- factor(DryBeandata$Class)
DryBeandata <- na.omit(DryBeandata)
DryBeandata[,1:16] <- as.data.frame(lapply(DryBeandata[,1:16], normalize))
str(DryBeandata)
result.DryBeandata <- Loop20_NB_FNB(DryBeandata)
#Iono:
Ionodata <- read.csv(choose.files())
Ionodata$Column35 <- factor(Ionodata$Column35)
Ionodata <- Ionodata[,c(-1,-2)]
Ionodata <- na.omit(Ionodata)
Ionodata[,1:32] <- as.data.frame(lapply(Ionodata[,1:32], normalize))
str(Ionodata)
result.Ionodata <- Loop20_NB_FNB(Ionodata)
#MGT:
```

```
MGTdata <- read.csv(choose.files())
MGTdata$class <- factor(MGTdata$class)
MGTdata <- na.omit(MGTdata)
MGTdata[,1:10] <- as.data.frame(lapply(MGTdata[,1:10], scale))
str(MGTdata)
result.MGTdata <- Loop20_NB_FNB(MGTdata)
#MHR:
MHRdata <- read.csv(choose.files())
MHRdata$RiskLevel <- factor(MHRdata$RiskLevel)
MHRdata <- na.omit(MHRdata)
MHRdata[,1:6] <- as.data.frame(lapply(MHRdata[,1:6], scale))
str(MHRdata)
result.MHRdata <- Loop20_NB_FNB(MHRdata)
#Raisin:
Raisindata <- read.csv(choose.files())
Raisindata$Class <- factor(Raisindata$Class)
Raisindata <- na.omit(Raisindata)
Raisindata[,1:7] <- as.data.frame(lapply(Raisindata[,1:7], scale))
str(Raisindata)
result.Raisindata <- Loop20_NB_FNB(Raisindata)
#Binary:
#4 vars., 200 size:
gnrt4_200 <- data.frame(v1 = rnorm(200), v2 = rnorm(200), v3 = rnorm(200), v4 = rnorm(200))
gnrt4_200$vclass <- (gnrt4_200[,1] > mean(gnrt4_200[,1]) | gnrt4_200[,2] > mean(gnrt4_200[,2])) &
(gnrt4_200[,3] > mean(gnrt4_200[,3]) | gnrt4_200[,4] > mean(gnrt4_200[,4]))
for (i in 1:length(gnrt4_200$vclass)) {
if (gnrt4_200$vclass[i] == TRUE) {gnrt4_200$vclass[i] <- "a"} else gnrt4_200$vclass[i] <- "b"}
gnrt4_200$vclass <- factor(gnrt4_200$vclass)
result.gnrt4_200 <- Loop20_NB_FNB(gnrt4_200)
#4 vars., 1000 size:
gnrt4_500 <- data.frame(v1 = rnorm(500), v2 = rnorm(500), v3 = rnorm(500), v4 = rnorm(500))
gnrt4_500$vclass <- (gnrt4_500[,1] > mean(gnrt4_500[,1]) | gnrt4_500[,2] > mean(gnrt4_500[,2])) &
(gnrt4_500[,3] > mean(gnrt4_500[,3]) | gnrt4_500[,4] > mean(gnrt4_500[,4]))
for (i in 1:length(gnrt4_500$vclass)) {
if (gnrt4_500$vclass[i] == TRUE) {gnrt4_500$vclass[i] <- "a"} else gnrt4_500$vclass[i] <- "b"}
gnrt4_500$vclass <- factor(gnrt4_500$vclass)
result.gnrt4_500 <- Loop20_NB_FNB(gnrt4_500)
#4 vars., 1000 size:
gnrt4_1000 <- data.frame(v1 = rnorm(1000), v2 = rnorm(1000), v3 = rnorm(1000), v4 = rnorm(1000))
gnrt4_1000$vclass <- (gnrt4_1000[,1] > mean(gnrt4_1000[,1]) | gnrt4_1000[,2] >
mean(gnrt4_1000[,2])) &
(gnrt4_1000[,3] > mean(gnrt4_1000[,3]) | gnrt4_1000[,4] > mean(gnrt4_1000[,4]))
for (i in 1:length(gnrt4_1000$vclass)) {
  if (gnrt4_1000$vclass[i] == TRUE) {gnrt4_1000$vclass[i] <- "a"} else gnrt4_1000$vclass[i] <- "b"}
gnrt4_1000$vclass <- factor(gnrt4_1000$vclass)
result.gnrt4_1000 <- Loop20_NB_FNB(gnrt4_1000)
```

b. صور لمخرجات تنفيذ التعليمات البرمجية:

```
> str(BCCdata)
'data.frame':   116 obs. of  10 variables:
 $ Age         : num  0.369 0.908 0.892 0.677 0.954 ...
 $ BMI         : num  0.254 0.115 0.235 0.148 0.136 ...
 $ Glucose     : num  0.0709 0.227 0.2199 0.1206 0.227 ...
 $ Insulin     : num  0.00491 0.01219 0.03687 0.01417 0.01994 ...
 $ HOMA        : num  0 0.00974 0.02206 0.00591 0.01375 ...
 $ Leptin      : num  0.0523 0.0527 0.1585 0.0648 0.0278 ...
 $ Adiponectin: num  0.2212 0.1037 0.571 0.1515 0.0869 ...
 $ Resistin    : num  0.0607 0.0108 0.0769 0.1211 0.0934 ...
 $ MCP.1       : num  0.225 0.256 0.308 0.534 0.441 ...
 $ Class_name : Factor w/ 2 levels "Healthy controls",..: 1 1 1 1 1 1 1 1 1 1 ...
> result.BCCdata$Stats
  Method Mean_Accuracy Sd_Accuracy Mean_RunTime Sd_RunTime
1     NB         0.6310      0.0717       0.0016     0.0007
2    FNB         0.7121      0.0897       0.0100     0.0017
> result.BCCdata$`T-test`

        Welch Two Sample t-test

data:  allacc[, 2] and allacc[, 1]
t = 3.1557, df = 36.248, p-value = 0.003216
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.02896974 0.13311026
sample estimates:
mean of x mean of y
  0.71208   0.63104
```

```
> str(DryBeandata)
'data.frame':   13611 obs. of  17 variables:
 $ Area           : num  0.0341 0.0355 0.0383 0.0409 0.0415 ...
 $ Perimeter      : num  0.0586 0.0776 0.068 0.0829 0.0653 ...
 $ MajorAxisLength: num  0.0443 0.0305 0.0526 0.0485 0.0329 ...
 $ MinorAxisLength: num  0.152 0.178 0.158 0.178 0.201 ...
 $ AspectRation   : num  0.1226 0.0516 0.1315 0.0916 0.0256 ...
 $ Eccentricity   : num  0.478 0.278 0.496 0.404 0.166 ...
 $ ConvexArea     : num  0.0331 0.035 0.0371 0.0414 0.0401 ...
 $ EquivDiameter  : num  0.0708 0.0736 0.0788 0.0839 0.0849 ...
 $ Extent         : num  0.671 0.736 0.717 0.731 0.701 ...
 $ Solidity       : num  0.923 0.872 0.932 0.762 0.95 ...
 $ roundness      : num  0.935 0.793 0.915 0.827 0.988 ...
 $ Compactness    : num  0.787 0.904 0.774 0.83 0.952 ...
 $ ShapeFactor1   : num  0.593 0.547 0.582 0.552 0.511 ...
 $ ShapeFactor2   : num  0.833 0.967 0.801 0.855 1 ...
 $ ShapeFactor3   : num  0.751 0.885 0.736 0.8 0.942 ...
 $ ShapeFactor4   : num  0.981 0.975 0.987 0.894 0.989 ...
 $ Class          : Factor w/ 7 levels "BARBUNYA","BOMBAY",..: 6 6 6 6 6 6 6 6 6 6 ...
> result.DryBeandata$Stats
  Method Mean_Accuracy Sd_Accuracy Mean_RunTime Sd_RunTime
1     NB         0.8976      0.0043       0.0411     0.0705
2    FNB         0.8998      0.0042       0.1471     0.0113
> result.DryBeandata$`T-test`

        Welch Two Sample t-test

data:  allacc[, 2] and allacc[, 1]
t = 1.7004, df = 37.986, p-value = 0.09722
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.0004315276  0.0049615276
sample estimates:
mean of x mean of y
 0.899825  0.897560
```

```
> str(Ionodata)
'data.frame':   351 obs. of  33 variables:
 $ Column3 : num  0.998 1 1 1 1 ...
 $ Column4 : num  0.471 0.406 0.483 0.274 0.488 ...
 $ Column5 : num  0.926 0.965 1 1 0.971 ...
 $ Column6 : num  0.512 0.319 0.502 1 0.533 ...
 $ Column7 : num  0.917 0.446 1 0.856 0.961 ...
 $ Column8 : num  0.311 0.032 0.44 0 0.384 ...
 $ Column9 : num  1 1 0.945 0.5 0.886 ...
 $ Column10: num  0.519 0.477 0.506 0.5 0.418 ...
 $ Column11: num  0.926 0.754 0.865 0.5 0.764 ...
 $ Column12: num  0.411 0.161 0.527 0.5 0.399 ...
 $ Column13: num  0.799 0.672 0.927 0.5 0.782 ...
 $ Column14: num  0.275 0.151 0.504 0.5 0.496 ...
 $ Column15: num  0.803 0.242 0.773 0 0.672 ...
 $ Column16: num  0.3089 0.0124 0.5015 0.5726 0.3627 ...
 $ Column17: num  0.922 0.527 0.919 0.77 0.765 ...
 $ Column18: num  0.307 0.189 0.432 0.303 0.391 ...
 $ Column19: num  0.791 0.666 0.878 0 0.726 ...
 $ Column20: num  0.339 0 0.457 0.228 0.411 ...
 $ Column21: num  0.785 0.434 0.854 0.15 0.53 ...
 $ Column22: num  0.352 0.273 0.362 1 0.322 ...
 $ Column23: num  0.685 0.41 0.717 0.5 0.512 ...
 $ Column24: num  0.263 0.321 0.44 0.5 0.236 ...
 $ Column25: num  0.784 0.398 0.788 1 0.516 ...
 $ Column26: num  0.244 0.367 0.299 0.953 0.174 ...
 $ Column27: num  0.705 0.398 0.795 0.758 0.566 ...
 $ Column28: num  0.269 0.408 0.389 1 0.234 ...
 $ Column29: num  0.606 0.405 0.716 1 0.512 ...
 $ Column30: num  0.33 0.442 0.413 0.4 0.189 ...
 $ Column31: num  0.711 0.417 0.802 0.628 0.471 ...
 $ Column32: num  0.228 0.469 0.379 1 0.202 ...
 $ Column33: num  0.593 0.431 0.78 0.338 0.477 ...
 $ Column34: num  0.273 0.488 0.309 1 0.172 ...
 $ Column35: Factor w/ 2 levels "b","g": 2 1 2 1 2 1 2 1 2 1 ...
> result.Ionodata$Stats
  Method Mean_Accuracy Sd_Accuracy Mean_RunTime Sd_RunTime
1     NB        0.8210      0.0344       0.0060     0.0017
2    FNB        0.9045      0.0262       0.0402     0.0105
> result.Ionodata$`T-test`

        Welch Two Sample t-test

data:  allacc[, 2] and allacc[, 1]
t = 8.6467, df = 35.5, p-value = 2.924e-10
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.06391698 0.10311302
sample estimates:
mean of x mean of y
 0.904545  0.821030
```

```
> str(MGTdata)
'data.frame':   19020 obs. of  11 variables:
 $ fLength : num  -0.577 -0.511 2.568 -0.695 0.517 ...
 $ fWidth  : num  -0.337 -0.57 6.206 -0.687 0.476 ...
 $ fSize   : num  -0.381 -0.649 2.616 -1.029 0.711 ...
 $ fConc   : num  0.0628 0.8204 -1.8758 1.282 -0.3475 ...
 $ fConc1  : num  -0.149 1.472 -1.773 1.607 -0.285 ...
 $ fAsym   : num  0.541 0.5169 2.0449 0.5328 -0.0202 ...
 $ fM3Long : num  0.225 0.26 -1.478 -0.334 0.353 ...
 $ fM3Trans: num  -0.406 -0.49 -2.183 -0.355 1.037 ...
 $ fAlpha  : num  0.477 -0.815 1.889 -0.659 -0.881 ...
 $ fDist   : num  -1.498 0.153 0.843 -1.031 2.176 ...
 $ class   : Factor w/ 2 levels "g","h": 1 1 1 1 1 1 1 1 1 1 ...
> result.MGTdata$Stats
  Method Mean_Accuracy Sd_Accuracy Mean_RunTime Sd_RunTime
1     NB        0.7283      0.0063       0.0246     0.0470
2    FNB        0.7870      0.0061       0.0996     0.0466
> result.MGTdata$`T-test`

        Welch Two Sample t-test

data:  allacc[, 2] and allacc[, 1]
t = 29.976, df = 37.966, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.05475429 0.06268571
sample estimates:
mean of x mean of y
 0.787005  0.728285
```

```
> str(MHRdata)
'data.frame':   1014 obs. of  7 variables:
 $ ï..Age     : num  -0.36156 0.38059 -0.0647 0.00951 0.38059 ...
 $ SystolicBP : num  0.913 1.456 -1.261 1.456 0.37 ...
 $ DiastolicBP: num  0.255 0.975 -0.465 0.615 -1.185 ...
 $ BS         : num  1.905 1.298 -0.22 -0.524 -0.797 ...
 $ BodyTemp   : num  -0.485 -0.485 0.973 -0.485 -0.485 ...
 $ HeartRate  : num  1.446 -0.532 0.704 -0.532 0.21 ...
 $ RiskLevel  : Factor w/ 3 levels "high risk","low risk",..: 1 1 1 1 2 1 3 1 3 1 ...
> result.MHRdata$Stats
  Method Mean_Accuracy Sd_Accuracy Mean_RunTime Sd_RunTime
1     NB        0.6000      0.0239       0.0021     0.0019
2    FNB        0.7079      0.0249       0.0120     0.0033
> result.MHRdata$`T-test`

        Welch Two Sample t-test

data:  allacc[, 2] and allacc[, 1]
t = 13.98, df = 37.94, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.09224883 0.12349117
sample estimates:
mean of x mean of y
 0.707865  0.599995
```

```
> str(Raisindata)
'data.frame':   900 obs. of  8 variables:
 $ Area           : num  -0.00718 -0.32404 0.07825 -1.07369 -0.21527 ...
 $ MajorAxisLength: num  0.0975 -0.2089 0.0977 -1.2444 -0.6786 ...
 $ MinorAxisLength: num  -0.0239 -0.2292 0.2369 -0.9148 0.7269 ...
 $ Eccentricity   : num  0.423 0.224 0.186 -1.069 -2.408 ...
 $ ConvexArea     : num  -0.0157 -0.3041 0.0621 -1.0756 -0.2385 ...
 $ Extent         : num  1.10613 -0.28762 -1.15761 0.00171 1.74429 ...
 $ Perimeter      : num  0.0662 -0.1612 0.1559 -1.1753 -0.3385 ...
 $ Class          : Factor w/ 2 levels "Besni","Kecimen": 2 2 2 2 2 2 2 2 2 2 ...
> result.Raisindata$Stats
  Method Mean_Accuracy Sd_Accuracy Mean_RunTime Sd_RunTime
1     NB        0.8373      0.0174       0.0026     0.0020
2    FNB        0.8544      0.0227       0.0099     0.0076
> result.Raisindata$`T-test`

        Welch Two Sample t-test

data:  allacc[, 2] and allacc[, 1]
t = 2.6772, df = 35.621, p-value = 0.01115
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.004141381 0.030058619
sample estimates:
mean of x mean of y
  0.85444   0.83734
```

```
> str(gnrt4_200)
'data.frame':   200 obs. of  5 variables:
 $ v1    : num  1.109 -0.646 -0.468 -0.493 -1.806 ...
 $ v2    : num  -2.303 0.581 -0.398 -0.401 -2.179 ...
 $ v3    : num  -0.686 -0.493 0.154 0.447 -0.638 ...
 $ v4    : num  0.544 -0.8 0.39 -1.412 0.715 ...
 $ vclass: Factor w/ 2 levels "a","b": 1 2 2 2 2 2 2 2 1 2 ...
> result.gnrt4_200$Stats
  Method Mean_Accuracy Sd_Accuracy Mean_RunTime Sd_RunTime
1     NB         0.799      0.0586       0.0010      5e-04
2    FNB         0.714      0.0592       0.0046      9e-04
> result.gnrt4_200$`T-test`

        Welch Two Sample t-test

data:  allacc[, 2] and allacc[, 1]
t = -4.5665, df = 37.996, p-value = 5.092e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.1226818 -0.0473182
sample estimates:
mean of x mean of y
    0.714     0.799
```

```
> str(gnrt4_500)
'data.frame':   500 obs. of  5 variables:
 $ v1    : num  0.836 -0.465 1.314 -0.462 -2.178 ...
 $ v2    : num  -1.029 0.932 -0.497 0.938 0.207 ...
 $ v3    : num  0.284 -0.758 0.158 0.13 0.381 ...
 $ v4    : num  -0.7372 1.3755 1.3694 -1.3462 -0.0223 ...
 $ vclass: Factor w/ 2 levels "a","b": 1 1 1 1 1 1 2 1 1 1 ...
> result.gnrt4_500$Stats
  Method Mean_Accuracy Sd_Accuracy Mean_RunTime Sd_RunTime
1     NB        0.7628      0.0390       0.0011     0.0007
2    FNB        0.7356      0.0261       0.0043     0.0019
> result.gnrt4_500$`T-test`

        Welch Two Sample t-test

data:  allacc[, 2] and allacc[, 1]
t = -2.5887, df = 33.188, p-value = 0.01419
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.048572338 -0.005827662
sample estimates:
mean of x mean of y
   0.7356    0.7628
```

```
> str(gnrt4_1000)
'data.frame':   1000 obs. of  5 variables:
 $ v1    : num  -0.0323 0.6956 2.0062 1.3766 1.0581 ...
 $ v2    : num  1.206 2.161 -1.229 1.173 -0.967 ...
 $ v3    : num  2.1406 1.2194 -0.0781 0.0249 0.8077 ...
 $ v4    : num  0.0137 0.4904 -1.0808 -0.1602 -0.1673 ...
 $ vclass: Factor w/ 2 levels "a","b": 1 1 2 1 1 1 2 2 2 2 ...
> result.gnrt4_1000$Stats
  Method Mean_Accuracy Sd_Accuracy Mean_RunTime Sd_RunTime
1     NB        0.7734      0.0236       0.0012     0.0004
2    FNB        0.7698      0.0317       0.0050     0.0013
> result.gnrt4_1000$`T-test`

        Welch Two Sample t-test

data:  allacc[, 2] and allacc[, 1]
t = -0.40685, df = 35.101, p-value = 0.6866
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.02156142  0.01436142
sample estimates:
mean of x mean of y
   0.7698    0.7734
```

9. المراجع:

1. Biau, G., & Devroye, L. (2015). Lectures on the Nearest Neighbor Method. Montreal: Canada. Springer. P:284.
2. De Mello, R., Ponti, M. (2018). A Practical Approach on the Statistical Learning Theory. Cham: Switzerland. Springer. P:373.
3. Devroye, L., Gyorfi, L., & Lugosi, G. (1996). A Probabilistic Theory of Pattern Recognition. New York: USA. Springer. P:631.
4. Fawcett, T. (2005 December). An introduction to ROC analysis. Pattern Recognition Letters. Vol: 27. P – P: 861 – 874. Elsevier Inc.
5. Gramacki, A. (2018). Nonparametric Kernel Density Estimation and Its Computational Aspects. Zielona Góra: Poland. Springer. P:197.
6. Hamel, L. (2009). Knowledge discovery with support vector machine. Oxford: UK. Wiley. P:267.
7. Hogg, R. V., Mckean, J. W., & Craig, A. T. (2018). Introduction to Mathematical Statistics. 8th Ed. Boston: USA. Pearson. P:762.
8. Izenman, A. (2008). Modern Multivariate Statistical Techniques. Philadelphia: USA. Springer. P:756.
9. John, G. H., & Langley, P. (2009). Estimating Continuous Distributions in Bayesian Classifiers. Stanford: USA.
10. Kulkarni, S., Harman, G. (2011). An Elementary Introduction to Statistical Learning Theory. New Jersey: USA. Wiley. P:221.
11. Murty, M. N., & Devi, V. S. (2012). Pattern Recognition - An Algorithmic Approach. London: England. Springer. P:276.
12. Pérez, A., Larrañaga, P., & Inza, I. (2009). Bayesian classifiers based on kernel density estimation: Flexible classifiers. International Journal of Approximate Reasoning. Vol: 50. P – P: 341 – 362. Elsevier Inc.
13. Scott, D. W. (1992). Multivariate Density Estimation. New York: USA. Wiley. P:329.