**MỤC LỤC**

[**Phần 1: Giới thiệu thuật toán Naive Bayes Classifier** 2](#_Toc77748975)

[**1.** **Cơ sở lý thuyết** 2](#_Toc77748976)

[**1.1 Định lý Bayes** 2](#_Toc77748977)

[**2.** **Naive Bayes Classifier (NBC)** 3](#_Toc77748978)

[**2.1 Các phân phối thường dùng** 3](#_Toc77748979)

[**2.2 Ứng dụng Naive Bayes Classifier** 4](#_Toc77748980)

[**Phần 2: Giải quyết bài toán với Naive Bayes Classifier** 5](#_Toc77748981)

[**1.** **Giới thiệu Python và các thư viện** 5](#_Toc77748982)

[**1.1 Ngôn ngữ lập trình Python** 5](#_Toc77748983)

[**1.2** **Các thư viện sử dụng** 5](#_Toc77748984)

[**2.** **Bài toán: “Liệu một người có kiếm được hơn 50.000$/năm không?”** 6](#_Toc77748985)

[**2.1. Xử lý dữ liệu** 6](#_Toc77748986)

[**2.2 Phân tích các loại biến** 8](#_Toc77748987)

[**2.3** **Xây dựng model** 12](#_Toc77748988)

[**2.4 Đánh giá mô hình** 17](#_Toc77748989)

[**2.5 Kết luận** 25](#_Toc77748990)

[**Phần 3: Đánh giá & Nhận xét** 26](#_Toc77748991)

[**1. Ưu điểm** 26](#_Toc77748992)

[**2. Nhược điểm** 26](#_Toc77748993)

[**3. Nhận xét** 26](#_Toc77748994)

[**Tài liệu tham khảo** 27](#_Toc77748995)

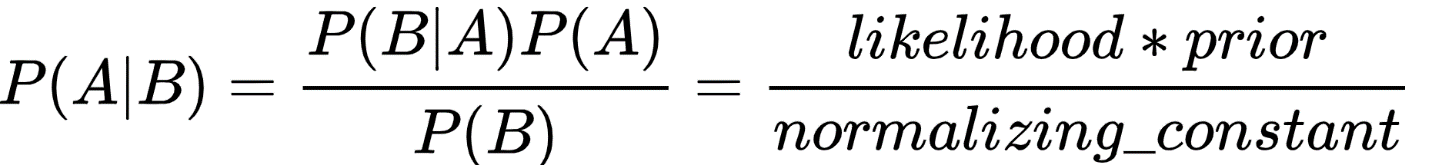
# **Phần 1: Giới thiệu thuật toán Naive Bayes Classifier**

1. **Cơ sở lý thuyết**

**1.1 Định lý Bayes**

Định lý Bayes cho phép tính xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra - “xác suất của A nếu có B” được ký hiệu là **P(A|B)**.

Công thức của định lý Bayes được phát biểu như sau:



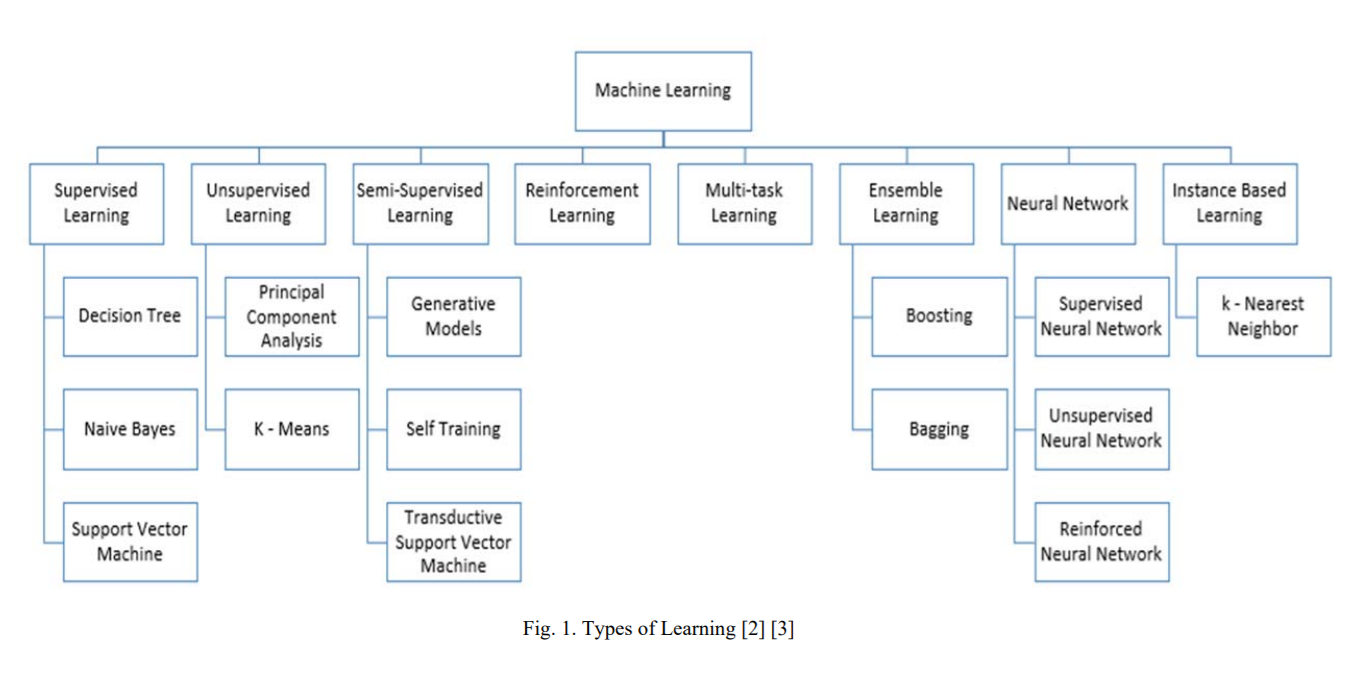
Theo định lý Bayes, xác suất xảy ra A khi biết B sẽ phụ thuộc vào 3 yếu tố:

* Xác suất xảy ra A của riêng nó, không quan tâm đến B. Kí hiệu là **P(A)** và đọc là xác suất của A. Đây được gọi là xác suất biên duyên hay xác suất tiên nghiệm, nó là “tiên nghiệm” theo nghĩa rằng nó không quan tâm đến bất kỳ thông tin nào về B.
* Xác suất xảy ra B của riêng nó, không quan tâm đến A. Kí hiệu là **P(B)** và đọc là “xác suất của B”. Đại lượng này còn gọi là hằng số chuẩn hóa (normalising constant), vì nó luôn giống nhau, không phụ thuộc vào sự kiện A đang muốn biết.
* Xác suất xảy ra B khi biết A xảy ra. Kí hiệu là **P(B|A)** và đọc là “xác suất của B nếu có A”. Đại lượng này gọi là khả năng (likelihood) xảy ra B khi biết A đã xảy ra. Chú ý không nhầm lẫn giữa khả năng xảy ra B khi biết A và xác suất xảy ra A khi biết B.

## **Naive Bayes Classifier (NBC)**

Trong thống kê, Naive Bayes Classifier là một họ của các "bộ phân loại theo xác suất " đơn giản dựa trên việc áp dụng định lý Bayes với các giả định về tính độc lập(naive) giữa các đối tượng. Chúng là một trong những mô hình mạng Bayes đơn giản nhất.

Trong học máy, Thuật toán phân lớp Naïve Bayes là một thuật toán Học có giám sát (Supervised Learning) đơn giản và mạnh mẽ cho việc phân loại.



Thuật toán phân lớp Naïve Bayes dựa trên việc áp dụng định lý Bayes với giả định về tính độc lập mạnh mẽ giữa các đối tượng. Thuật toán Phân lớp Naïve Bayes tạo ra kết quả tốt khi chúng ta sử dụng nó để phân tích những dữ liệu văn bản (như Xử lý ngôn ngữ tự nhiên, …).

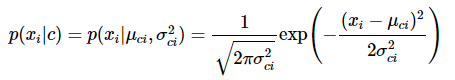
Mô hình Naïve Bayes còn được gọi là Bayes đơn giản hoặc Bayes độc lập. Tất cả những cái tên này đề cập đến ứng dụng của định lý Bayes trong quy tắc quyết định của trình phân loại.

### **2.1 Các phân phối thường dùng**

**2.1.1 Gaussian Naive Bayes**

Mô hình sử dụng chủ yếu cho loại dữ liệu có thành phần là các biến liên tục.

Với mỗi chiều dữ liệu **i** và một class **c**, **xi** tuân theo một phân phối chuẩn có kỳ vọng **µci** và phương sai **σ2ci**. Đây chính là cách tính của thư viện ***sklearn*** (trong Python)



* + 1. **Multinomial Naive Bayes**

Với mô hình Naïve Bayes đa thức, các mẫu (vectơ đặc trưng) đại diện cho các tần số mà các sự kiện nhất định đã được tạo ra bởi một đa thức (p1,..., Pn) trong đó pi là xác suất xảy ra sự kiện i. Thuật toán Naïve Bayes đa thức được ưu tiên sử dụng trên dữ liệu được phân phối đa thức. Nó là một trong những thuật toán tiêu chuẩn được sử dụng trong phân loại phân loại văn bản.

* + 1. **Bernoilli Naïve Bayes**

Mô hình này được áp dụng cho các loại dữ liệu mà mỗi thành phần là một giá trị binary – bằng 0 hay 1. Ví dụ: cũng với loại văn bản nhưng thay vì đếm tổng số lần xuất hiện của một từ trong văn bản, ta chỉ cần quan tâm từ đó có xuất hiện hay không thay vì tần suất xuất hiện.

Khi đó, p(xi|c) được tính bằng:



Với p(i|c) có thể được hiểu là xác suất từ thứ i xuất hiện trong các văn bản class c.

### **2.2 Ứng dụng Naive Bayes Classifier**

* **Dự đoán theo thời gian thực:** mô hình phân lớp NB chạy khá nhanh nên nó thích hợp ứng dụng nhiều vào các ứng dụng theo thời gian thực, như hệ thống cảnh báo, các hệ thống trading …
* **Dự đoán đa lớp:**Bayes mở rộng có thể ứng dụng vào các loại ứng dụng có thể dự đoán nhiều giả thuyết mục tiêu (ứng dụng đa dự đoán)
* **Phân loại văn bản:**NBC cũng đặc biệt phù hợp cho các hệ thống phân loại văn bản hoặc ngôn ngữ tự nhiên bởi tính chính xác lớn hơn các thuật toán khác.
* **Hệ thống gợi ý:**Naive Bayes Classifier cũng được sử dụng để xây dựng hệ thống gợi ý, ví dụ như xuất hiện các quảng cáo mà người dùng đang quan tâm nhiều từ việc phân tích thói quen sử dụng internet của người dùng, hoặc đưa ra gợi ý các bài hát tiếp theo mà người dùng sẽ thích trong một ứng dụng nghe nhạc (Sportify, AppleMusic,…)

# **Phần 2: Giải quyết bài toán với Naive Bayes Classifier**

1. **Giới thiệu Python và các thư viện**

**1.1 Ngôn ngữ lập trình Python**

**Python** là một ngôn ngữ lập trình bậc cao phổ biến, được tạo ra bởi **Guido van Rossum** và phát hành vào **năm 1991**.

Python là một trong những ngôn ngữ ứng dụng tốt vào lĩnh vực Trí tuệ nhân tạo và Khoa học dữ liệu

Ưu điểm khi sử dụng Python:

* Python hoạt động trên các nền tảng khác nhau (Windows, Mac, Linux, Raspberry Pi, v.v.).
* Python có cú pháp đơn giản tương tự như ngôn ngữ tiếng Anh.
* Python có cú pháp cho phép các nhà phát triển viết chương trình với ít dòng hơn một số ngôn ngữ lập trình khác.
* Python chạy trên một hệ thống thông dịch, có nghĩa là mã có thể được thực thi ngay sau khi nó được viết. Điều này có nghĩa là việc tạo mẫu có thể rất nhanh chóng.
* Python có thể được xử lý theo hướng thủ tục hoặc hướng đối tượng.

Phiên bản sử dụng trong bài: **python 3.9** chạy trên **JupyterNotebook**

* 1. **Các thư viện sử dụng**

**Numpy** (Numeric Python): là một thư viện toán học rât phổ biến và mạnh mẽ của Python. NumPy được trang bị các hàm số đã được tối ưu, cho phép làm việc hiệu quả với ma trận và mảng với tốc độ xử lý nhanh hơn nhiều lần.

**Pandas** là một thư viện sử dụng 2 loại dữ liệu chính là Series và DataFrame, cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này giúp làm việc với dữ liệu “quan hệ” hoặc “có nhãn” dễ dàng và trực quan.

**Scikit-learn** là một trong những thư viên tốt cung cấp các hàm về xử lý, sử dụng các thuật toán thống kê cũng như các mô hình học máy trong python.

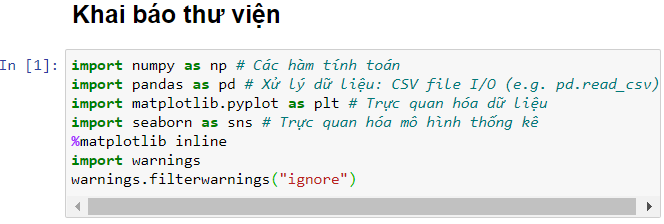
1. **Bài toán:   
   “Liệu một người có kiếm được hơn 50.000$/năm không?”**

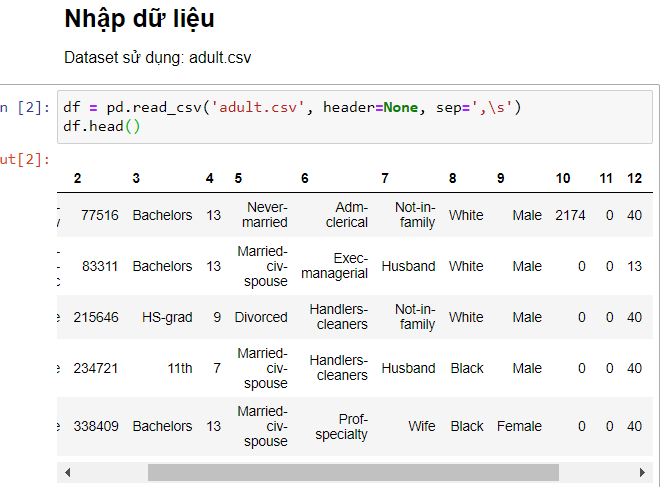
Trong bài toán này: Triển khai thuật toán Phân loại Naive Bayes với Python và Scikit-Learn để xây dựng Bộ phân loại Naive Bayes dự đoán “liệu một người có kiếm được hơn 50.000$/năm không? hay không.?”

Dataset: <https://www.kaggle.com/qizarafzaal/adult-dataset>

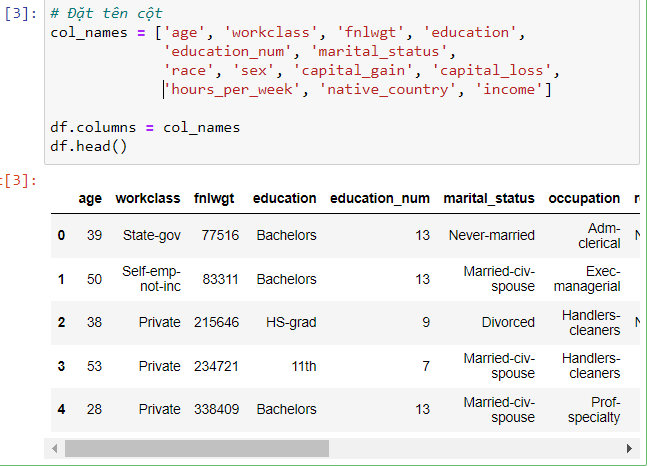
Source Code: <https://github.com/congltk1234/NhapMonKHDL/blob/main/%C4%90%E1%BB%93%20%C3%A1n%20NMKHDL/NaiveBayesClassifier.ipynb>

### **2.1. Xử lý dữ liệu**

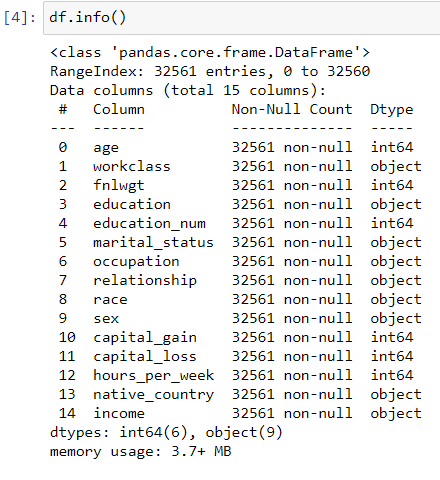




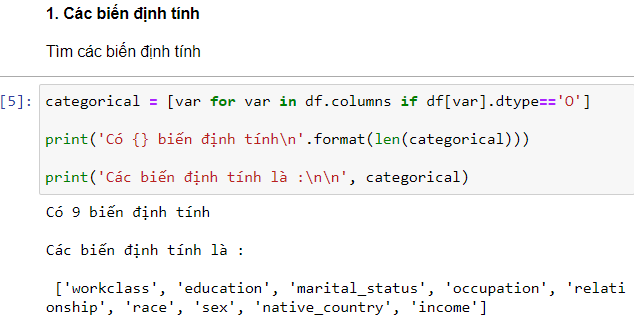
Chúng ta có thể thấy rằng tập dữ liệu không có tên cột thích hợp. Các cột chỉ được gắn nhãn là: 0,1,2 .... **=> Đặt lại tên cột**

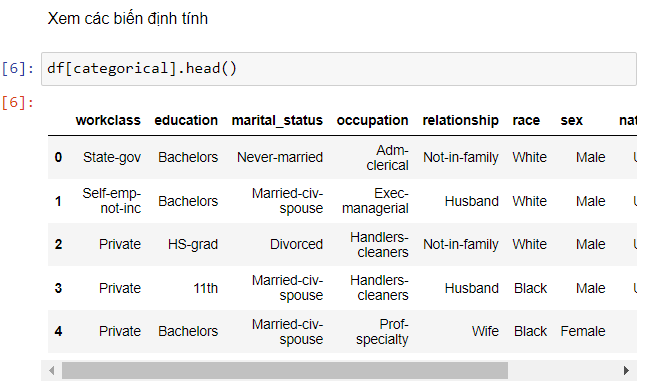


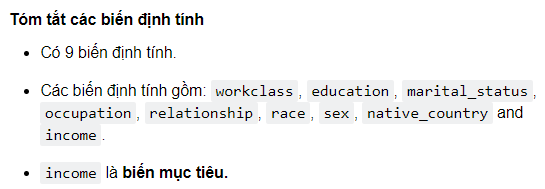
Xem tóm tắt của tập dữ liệu

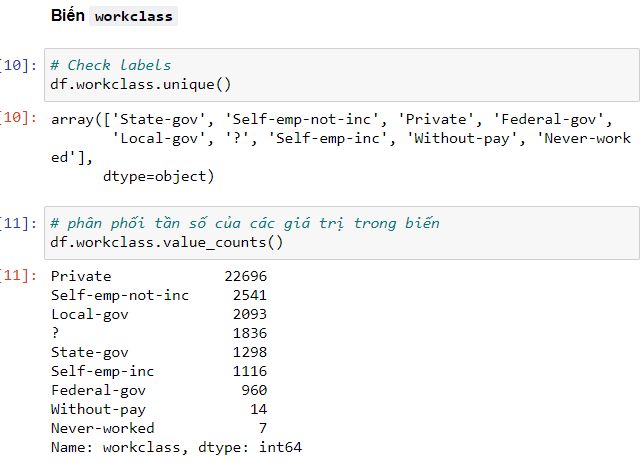


### **2.2 Phân tích các loại biến**

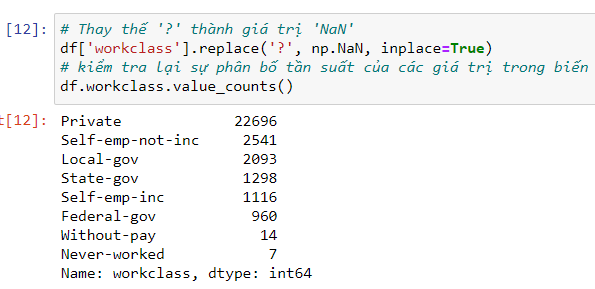






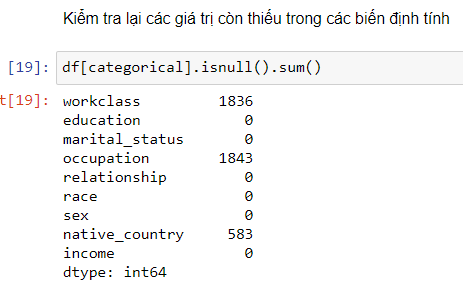


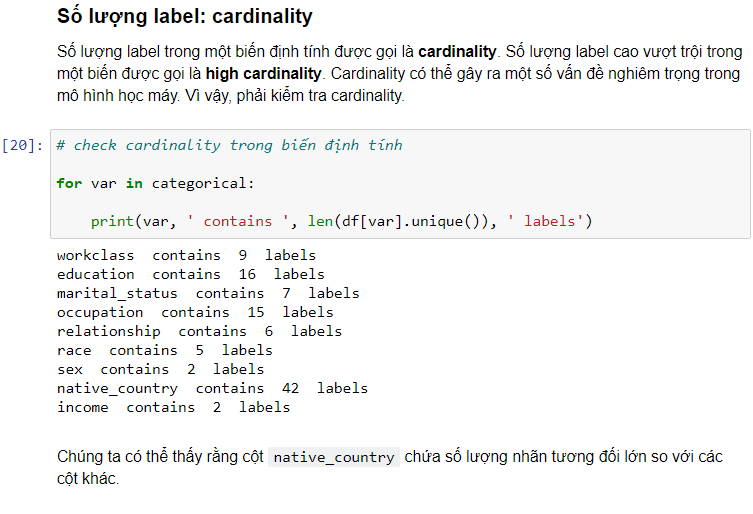
Có 1836 giá trị **?** trong **workclass**

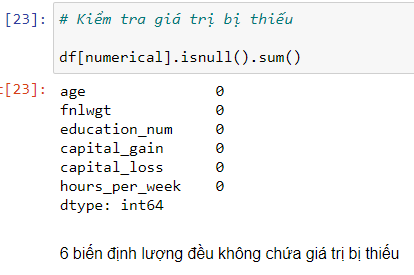
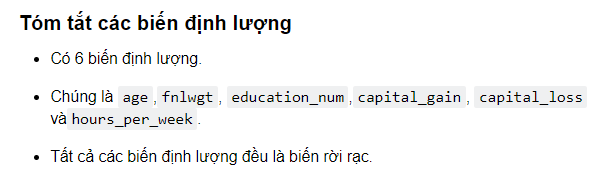
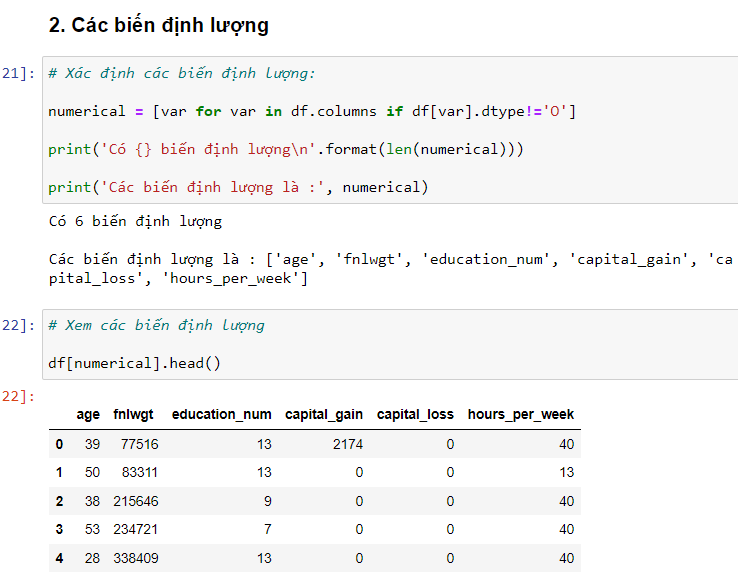


**Nhận thấy:** Các giá trị **?** đã bị thay thế

Thao tác tương tự với biến **occupation** và biến **native\_country**

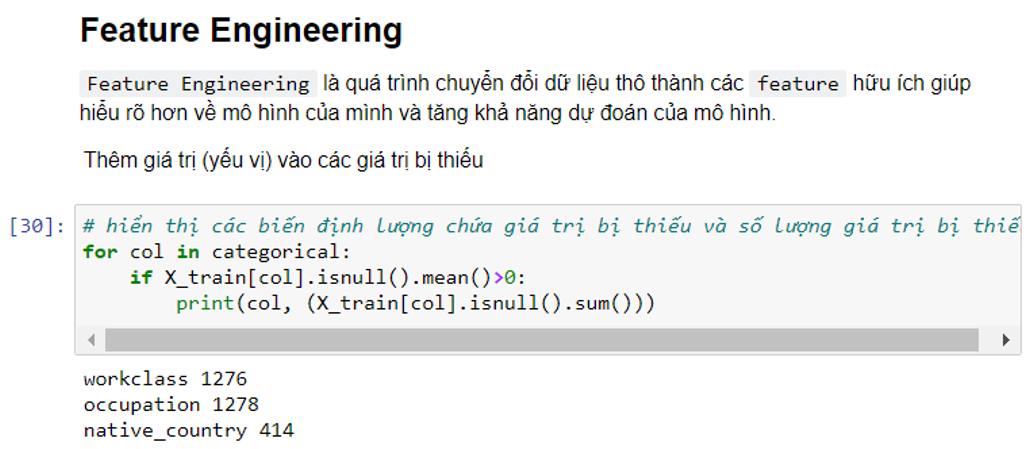
Bây giờ, chúng ta có thể thấy rằng biến  **workclass, Occupation** và **native\_country** chứa các giá trị bị thiếu.

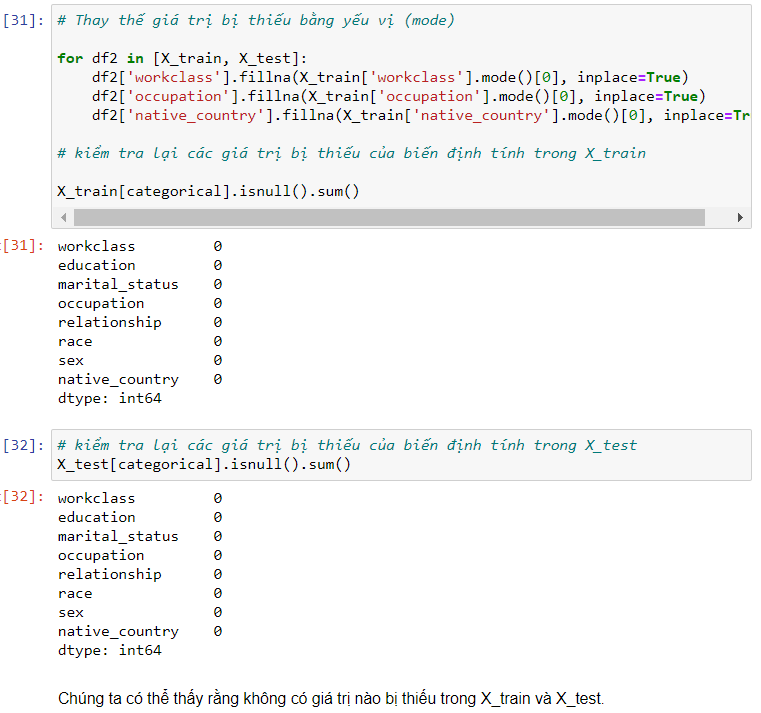




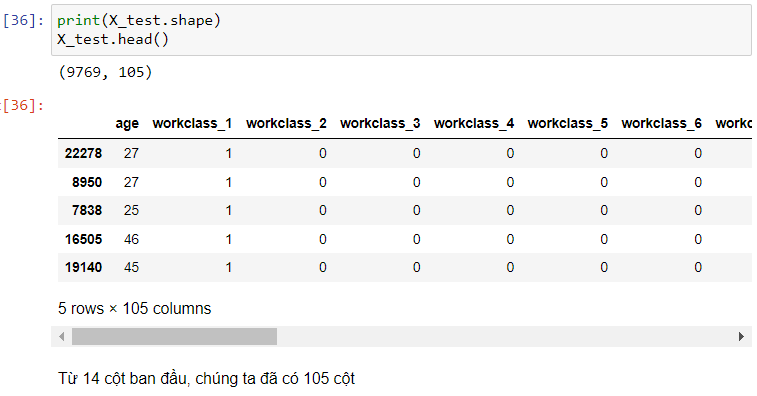
* 1. **Xây dựng model**

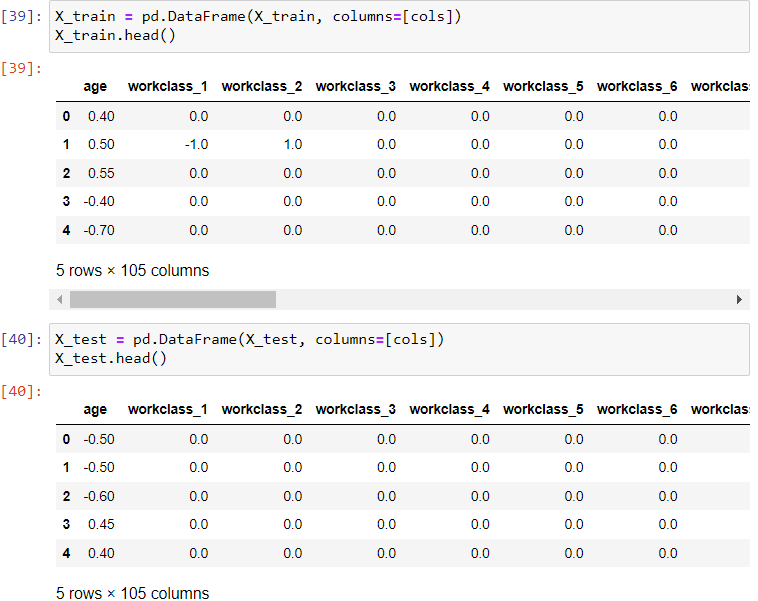
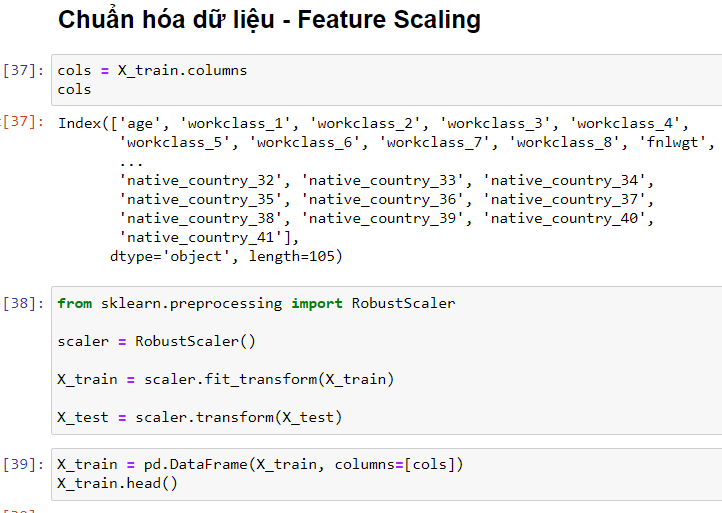


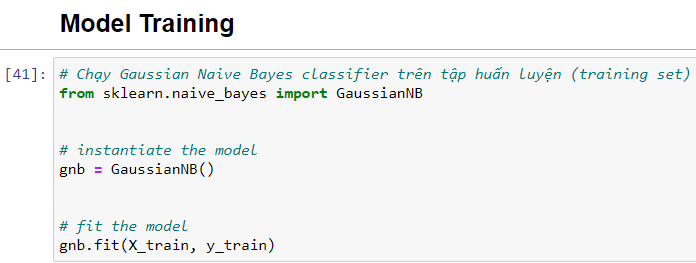




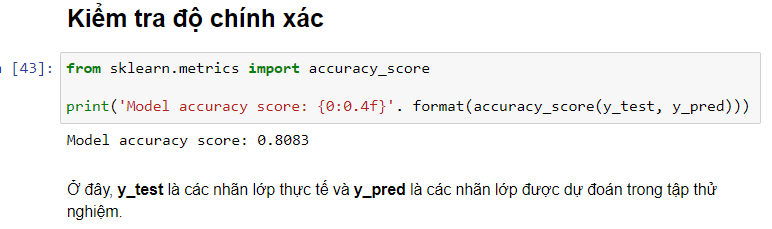


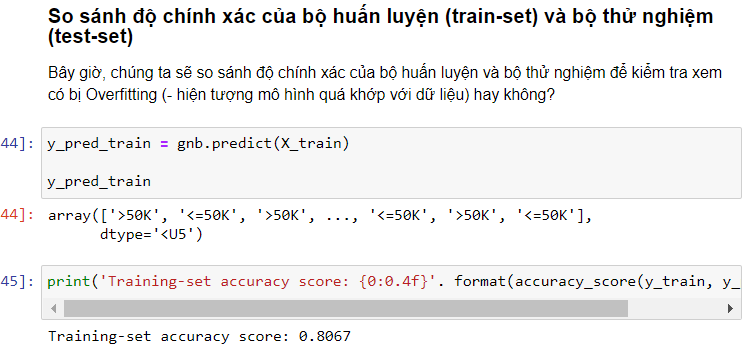


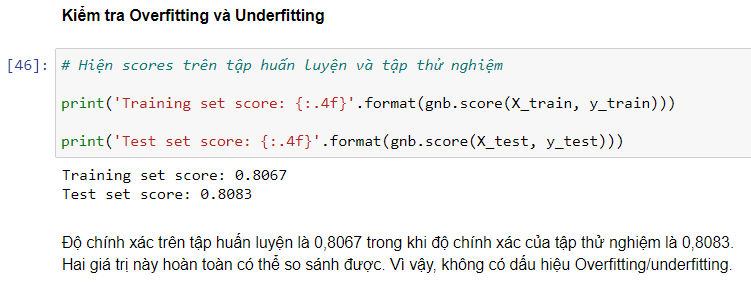


Bây giờ chúng ta đã có tập dữ liệu **X\_train** sau khi xử lý đã sẵn sàng được đưa vào bộ phân loại **Gaussian Naive Bayes.**

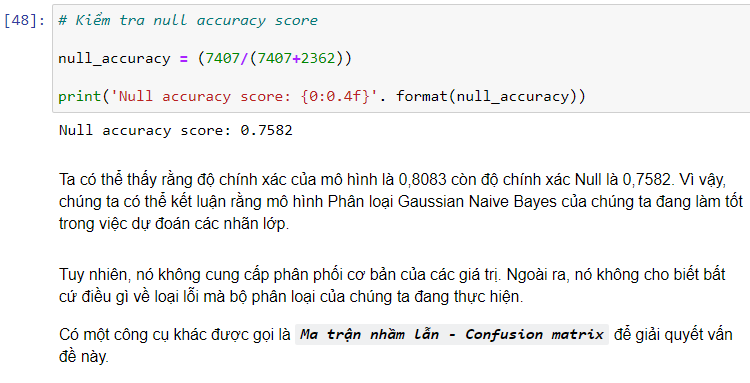
### **2.4 Đánh giá mô hình**

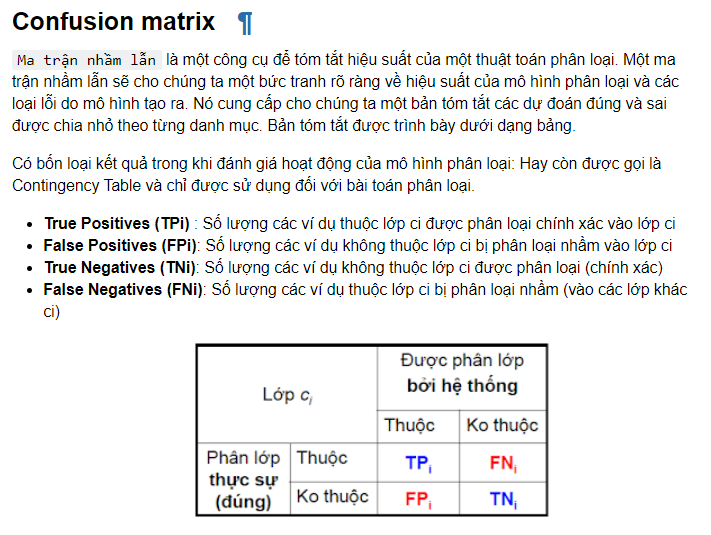


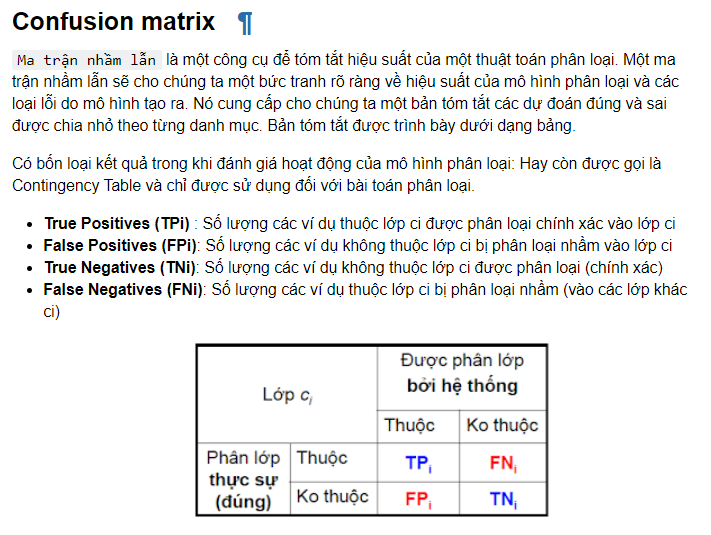


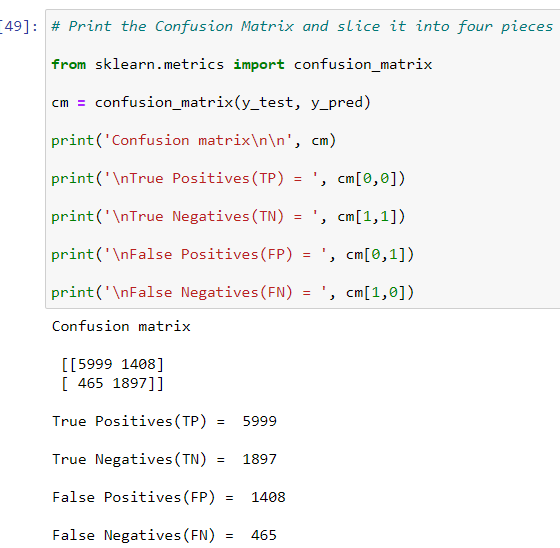


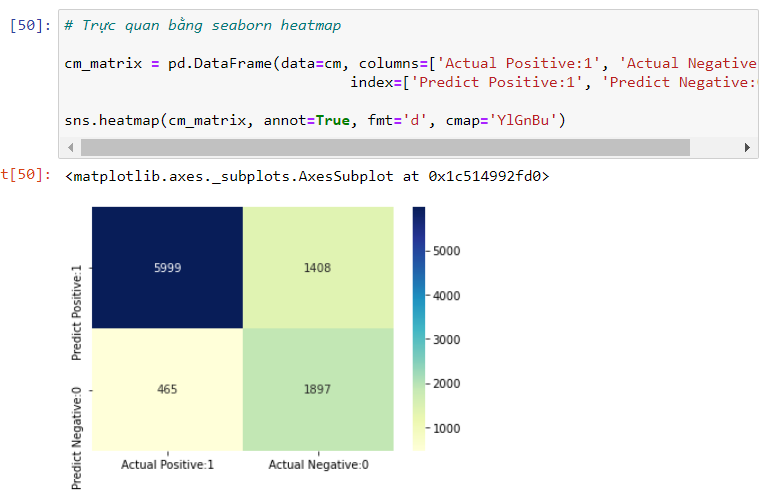


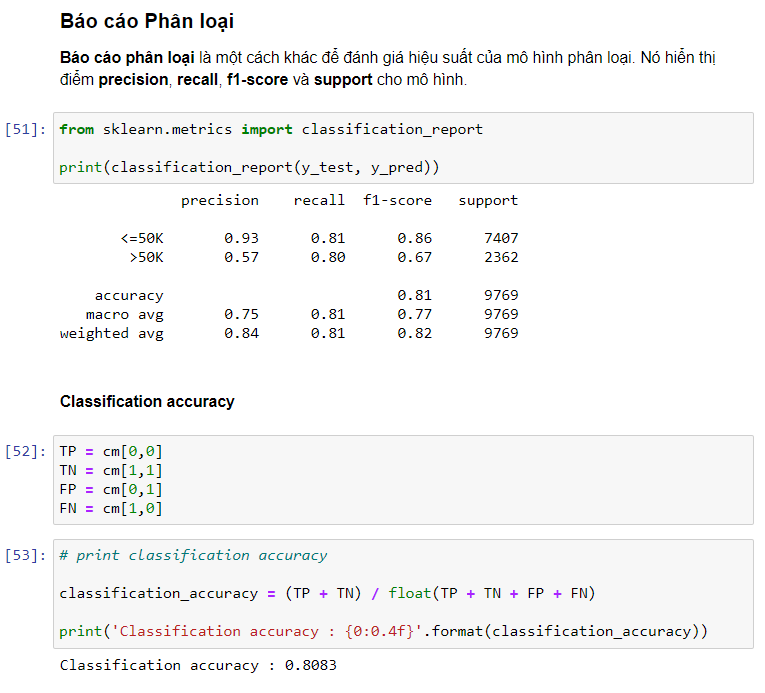










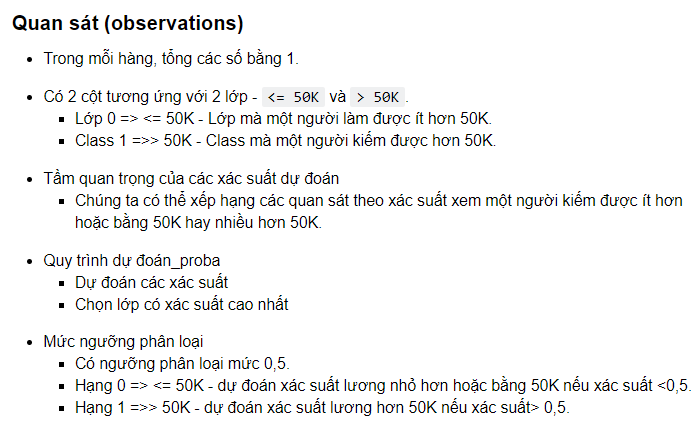
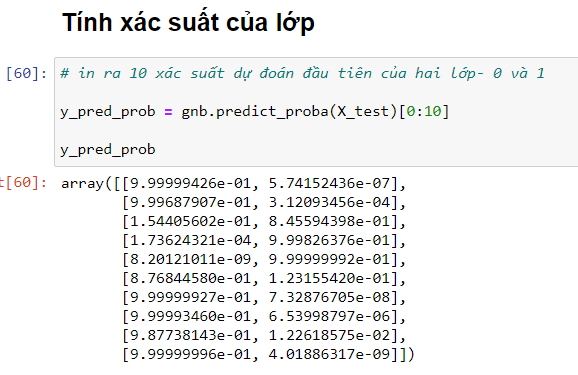


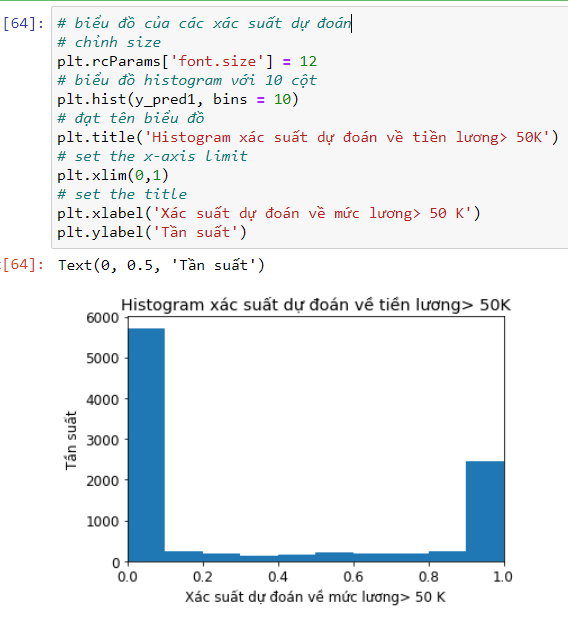
**Precision** có thể được định nghĩa là tỷ lệ phần trăm kết quả tích cực được dự đoán chính xác trong số tất cả các kết quả **tích cực được dự đoán.** Nó có thể được cho dưới dạng tỷ lệ giữa số (TP) trên tổng số (TP + FP).

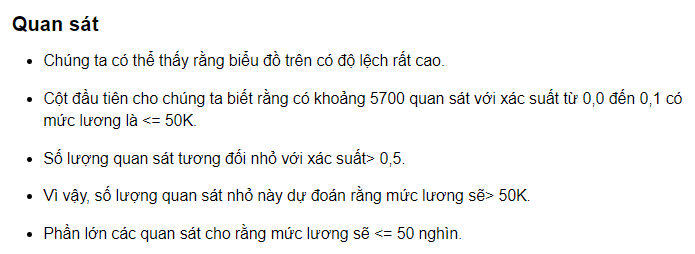
**Recall** có thể được định nghĩa là tỷ lệ phần trăm kết quả tích cực được dự đoán chính xác trong số tất cả các kết quả **tích cực thực tế**. Nó có thể được cho là tỷ lệ giữa (TP) trên tổng (TP + FN).

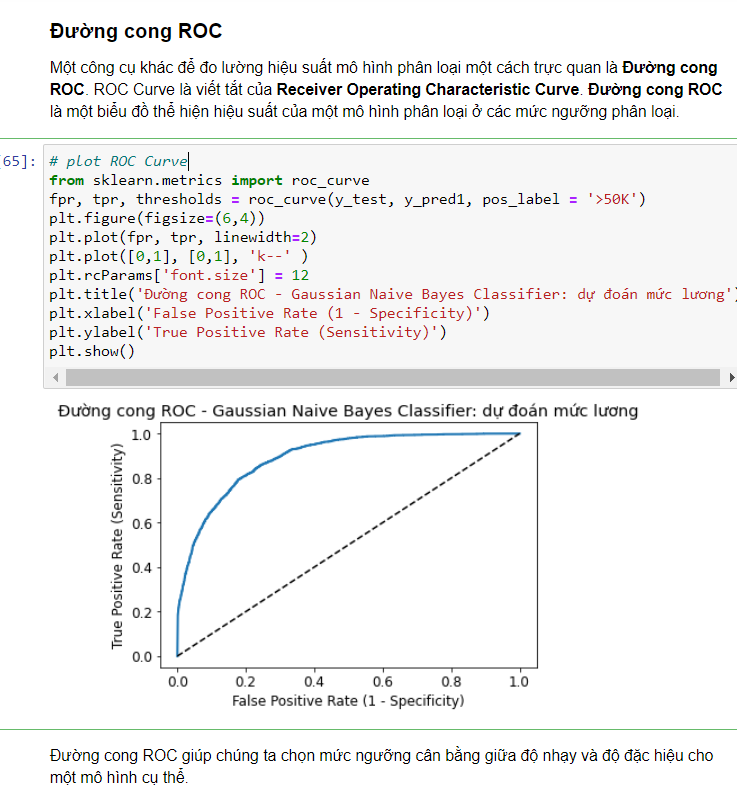
**f1-score** tốt nhất có thể là 1,0 và điểm kém nhất là 0,0. **F1-score** là **harmonic mean** của **precision** và **recall**. Giá trị trung bình có trọng số của **f1-score** nên được sử dụng để so sánh các mô hình phân loại, không phải là độ chính xác toàn cục.

**support** là số lần xuất hiện thực tế của lớp trong tập dữ liệu của chúng tôi.



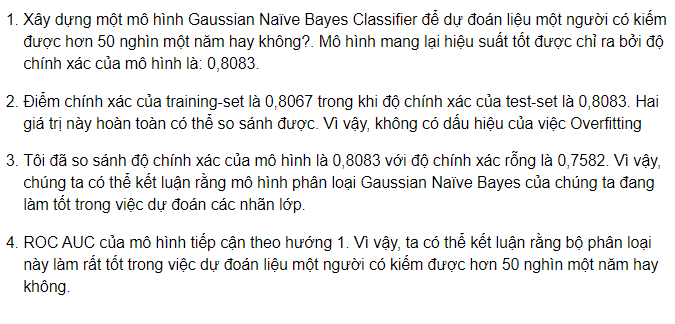








### **2.5 Kết luận**



# **Phần 3: Đánh giá & Nhận xét**

## **1. Ưu điểm**

* Giả định độc lập: hoạt động đủ tốt để giải quyết nhiều bài toán phân lớp đơn giản.
* Tốt khi có sự chệnh lệch số lượng giữa các lớp phân loại.
* Thuật toán Naive Bayes là một thuật toán nhanh, có khả năng mở rộng cao.
* Naive Bayes có thể được sử dụng để phân loại Binary và Multiclass. Nó cung cấp các loại Thuật toán Naive Bayes khác nhau như Gaussian, Multinomial, Bernoulli.
* Nó có thể dễ dàng train một model trên tập dữ liệu nhỏ

## **2. Nhược điểm**

* Giả định độc lập (ưu điểm cũng chính là nhược điểm)  
  hầu hết các trường hợp thực tế trong đó có các thuộc tính trong các đối tượng thường phụ thuộc lẫn nhau.
* Mô hình không được huẩn luyện bằng phượng pháp tối ưu mạnh và chặt chẽ. Tham số của mô hình là các ước lượng xác suất điều kiện đơn lẻ. Không tính đến sự tương tác giữa các ước lượng này.
* Dễ bị overfit nếu không cẩn thận

## **3. Nhận xét**

Mô hình Naive Bayes là mô hình phân lớp đơn giản dễ cài đặt, có tốc độ xử lý nhanh. Tuy nhiên có nhược điểm lớn là yêu cầu các đặc trưng đầu vào phải độc lập, mà điều này khó xảy ra trong thực tế làm giảm chất lượng của mô hình. Thuật toán này thường được sử dụng trong phân tích sắc thái, lọc thư rác, recommendation systems, ...

* NBC có thể hoạt động với các feature vector mà một phần là liên tục (sử dụng Gaussian Naive Bayes), phần còn lại ở dạng rời rạc (sử dụng Multinomial hoặc Bernoulli).
* Naive Bayes Classifiers (NBC) thường được sử dụng trong các bài toán Text Classification.
* NBC có thời gian training và test rất nhanh. Điều này có được là do giả sử về tính độc lập giữa các thành phần, nếu biết class.

# **Tài liệu tham khảo**

<https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/>

<https://viblo.asia/p/thuat-toan-phan-lop-naive-bayes-924lJWPm5PM>

<https://dataaspirant.com/naive-bayes-classifier-machine-learning/>

<https://machinelearningcoban.com/2017/08/08/nbc/>

Và ngoài ra còn một số trang QnA khác như Stackoverflow, Kaggle, …

Các Thư viện :

<https://pandas.pydata.org/>

<https://numpy.org/>

<https://scikit-learn.org/stable/modules/naive_bayes.html>

Dataset: <https://www.kaggle.com/qizarafzaal/adult-dataset>

Source Code:

<https://github.com/congltk1234/NhapMonKHDL/blob/main/%C4%90%E1%BB%93%20%C3%A1n%20NMKHDL/NaiveBayesClassifier.ipynb>