**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

****

**BÁO CÁO BÀI TẬP LỚN: DỰ ĐOÁN HÀNH VI TIÊU DÙNG BẰNG THUẬT TOÁN NAIVE BAYES**

**Sinh viên:** Nguyễn Việt Phương (24022431)

Lưu Xuân Tân (24022447)

Lê Nho Minh Hiếu (24022328)

**MỤC LỤC**

1. Khái quát về bài tập lớn:
2. Quyết định chọn dự án của nhóm
3. Sơ lược về dự án
4. Thu thập dữ liệu:
5. Giới thiệu về tệp dữ liệu sử dụng
6. Nhận xét ban đầu
7. Làm sạch và tiền xử lí dữ liệu:

1. Xử lí giá trị thiếu (missing values)

2. Tạo đặc trưng mới (feature engineering)

3. Loại bỏ trùng lặp (duplicate records)

4. Xử lí ngoại lệ (outliers)

1. Phân bổ tệp dữ liệu huấn luyện – kiểm thử:
2. Tạo biến mục tiêu
3. Lựa chọn đặc trưng
4. Phân chia dữ liệu
5. Mã hóa và chuẩn hóa:
6. Encoding
7. Chuẩn hóa dữ liệu
8. Cân bằng dữ liệu bằng kĩ thuật SMOTE
9. Huấn luyện mô hình Gaussian Naive Bayes:
10. So sánh hiệu quả mô hình:
11. Kết luận:
12. **Khái quát về bài tập lớn:**
13. **Bối cảnh triển khai:**

Trong môi trường thương mại điện tử cạnh tranh khốc liệt, việc hiểu rõ hành vi của người dùng là yếu tố sống còn giúp doanh nghiệp tối ưu hóa chiến lược kinh doanh — từ quảng cáo đến quản lý sản phẩm. Dự án này nhằm xây dựng một **mô hình học máy** có khả năng **dự đoán hành vi mua hàng** của khách hàng dựa trên dữ liệu tương tác của họ trên nền tảng.

1. **Sơ lược về ý tưởng:**

Trong bài toán dự đoán hành vi, ta có thể coi các hành vi của khách hàng cũng giống như một bài toán phân lớp nhị phân:

* Lớp 0: Người dùng không thực hiện hành vi mua hàng
* Lớp 1: Người dùng thực hiện hành vi mua hàng

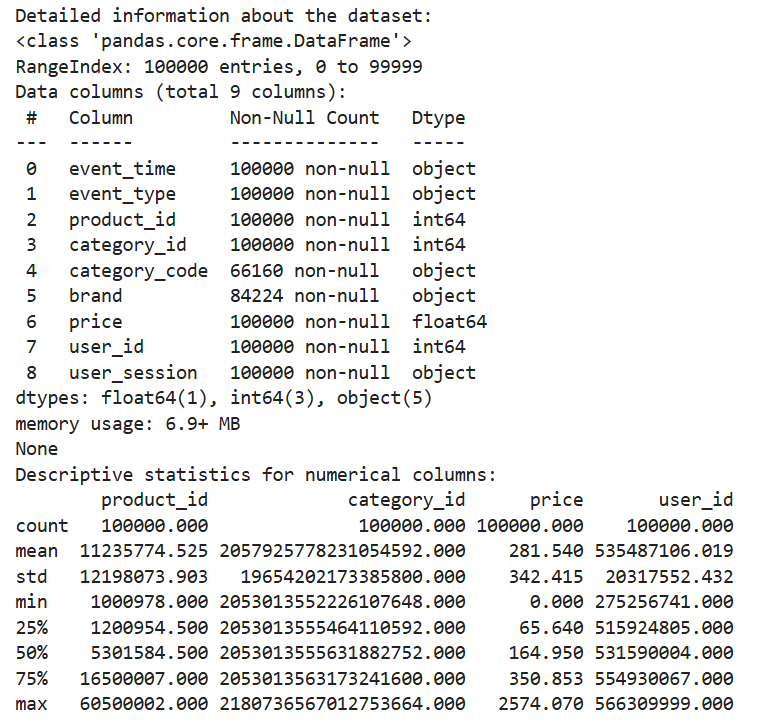
Yêu cầu bài toán cho thấy, việc sử dụng Naive Bayes là hoàn toàn hợp lí. Mô hình Naive Bayes được đánh giá cao trong các bài toán phân loại, đặc biệt là khi dữ liệu bao gồm các thuộc tinh dạng phân loại như giới tinh, độ tuổi hay nhóm sản phẩm mà khách hàng quan tâm. Bản chất của mô hình này là giả định tính độc lập giữa các đặc trưng đầu vào, từ đó đơn giản hóa việc tinh toán xác suất, giúp quá trình huấn luyện trở nên nhanh chóng và tối ưu tài nguyên. Nhờ đó, Naïve Bayes đặc biệt phù hợp với các hệ thống marketing đòi hỏi tốc độ phản hồi tức thì, cùng với việc hoạt động hiệu quả ngay cả khi lượng dữ liệu đầu vào không lớn.

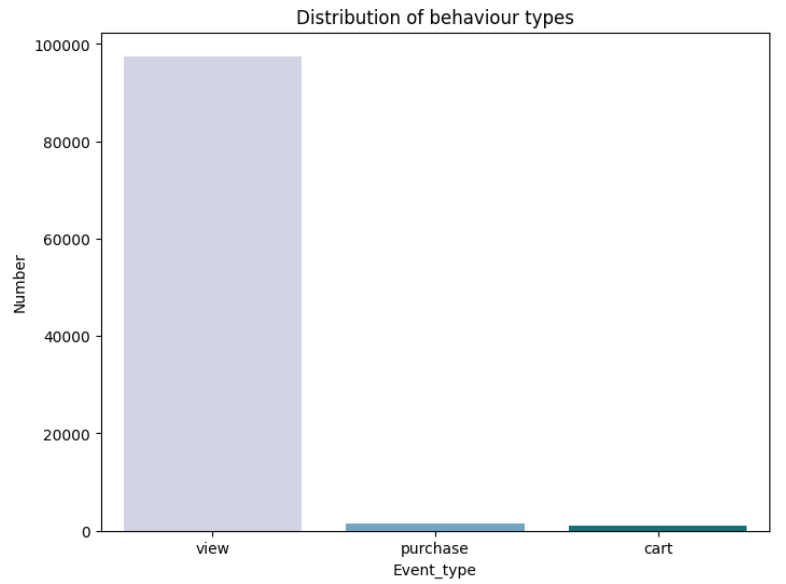
Tuy nhiên, một thách thức không nhỏ đặt ra là sự mất cân bằng dữ liệu – tình huống thường gặp khi số người mua hang thực tế chỉ chiếm một phần nhỏ so với số lượng người chỉ truy cập và xem qua sản phẩm. Sự mất cân bằng này có nguy cơ làm cho mô hình học lệch, đánh giá sai tầm quan trọng của lớp thiểu số, tức người mua hang. Để khắc phục tình trạng này, nhóm đề xuất sử dụng kĩ thuật Over-Sampling bằng phương pháp SMOTE (Synthetic Minority Over-Sampling Technique). Phương pháp này có khả năng tạo ra them các mẫu dữ liệu cho lớp thiểu số, giúp cân bằng tỉ lệ giữa hai lớp trong tập huấn luyện. Nhờ đó, mô hình Naïve Bayes có thể học được nhiều đặc trưng hơn từ nhóm khách hang thực hiện hành vi mua hàng, qua đó nâng cao độ chính xác trong dự đoán hành vi tiêu dùng thực tế.

1. **Thu thập dữ liệu:**
2. **Giới thiệu về tệp dữ liệu sử dụng:**

Trong dự án này, nhóm đã tiến hành thu thập dữ liệu từ một tệp có sẵn 2019\_Nov\_csv (github), tệp này gồm **100.000 bản** ghi giao dịch được thu thập trong năm 2019. Đây là dữ liệu hành vi người dung trên một nền tảng thương mại điện tử, phản ánh các hoạt động như:

* *event\_time:* Thời điểm xảy ra hành vi, giúp phân tích xu hướng hoạt động của người dùng theo thời gian
* *event\_type:* Loại hành vi của người dùng (view, cart, purchase)
* *product\_id:* Mã định danh của sản phẩm được tương tác
* *category\_code:* Danh mục/Nhóm sản phẩm tương ứng
* *brand:* Thương hiệu của sản phẩm
* *price:* Giá sản phẩm tính tai thời điểm người dùng truy cập
* *user\_id:* Mã người dùng thực hiện hành vi, đi kèm với một số trường bổ sung liên quan đến sản phẩm và người dùng (nếu có)





1. **Nhận xét ban đầu:**

Thông qua quá trình khảo sát ban đầu, nhóm nhận định như sau:

* Hành vi *‘view’* chiếm tỉ lệ áp đảo so với các hành vi khác, trong khi hành vi *‘purchase’* chiếm tỉ lệ rất thấp. Điều này phản ánh thực tế hành vi người tiêu dùng, khi phần lớn người dùng chỉ xem qua sản phẩm mà không thực hiện giao dịch mua.
* Các trường *‘category\_code’* và *‘brand’* có nhiều giá trị khuyết, có thể gây khó khăn cho quá trình huấn luyện mô hình và phân tích sâu theo nhóm sản phẩm.
* Trường *‘price’* có phân phối lệch, biểu thị sự chênh lệch lớn giữa các mức giá sản phẩm, đồng thời chứa nhiều giá trị ngoại lai (outliers), quá trình xử lí sẽ được mô tả rõ hơn trong phần sau của bài báo cáo.

1. **Làm sạch và tiền xử lí dữ liệu:**
2. **Xử lí giá trị thiếu (Missing Values):**

Sau quá trình quan sát, nhóm nhận thấy các cột ***‘category\_code’, ‘brand’*** và ***‘price’*** là những trường có tỉ lệ giá trị bị thiếu cao. Việc giữ nguyên các giá trị này có thể khiến mô hình khó học được mối quan hệ giữa các biến, do đó cần tiến hành xử lý như sau:

* Đối với ***‘price’***: Các giá trị bị thiếu được thay thế bằng median của toàn bộ cột đó, dùng median thay vì mean sẽ giúp ta giảm ảnh hưởng của outliers, đảm bảo độ ổn định và phản ánh đúng xu hướng giá chung.
* Đối với ***‘category\_code’*** và ***‘brand’***: Các giá trị bị thiếu được điền bằng chuỗi ‘unknown’ nhằm duy trì kích thước dữ liệu gốc mà không bỏ dataset. Cách tiếp cận này đảm bảo mô hình có thể nhận diện những mẫu dữ liệu không rõ nguồn gốc sản phẩm hay thương hiệu mà không làm mất thông tin.

1. **Tạo đặc trưng mới (Feature Engineering):**

Nhóm đã tiến hành trích xuất them các đặc trưng mới từ trường ***‘event\_time’*** nhằm cải thiện khả năng hiểu hành vi người dùng theo yếu tố thời gian của mô hình, cụ thể:

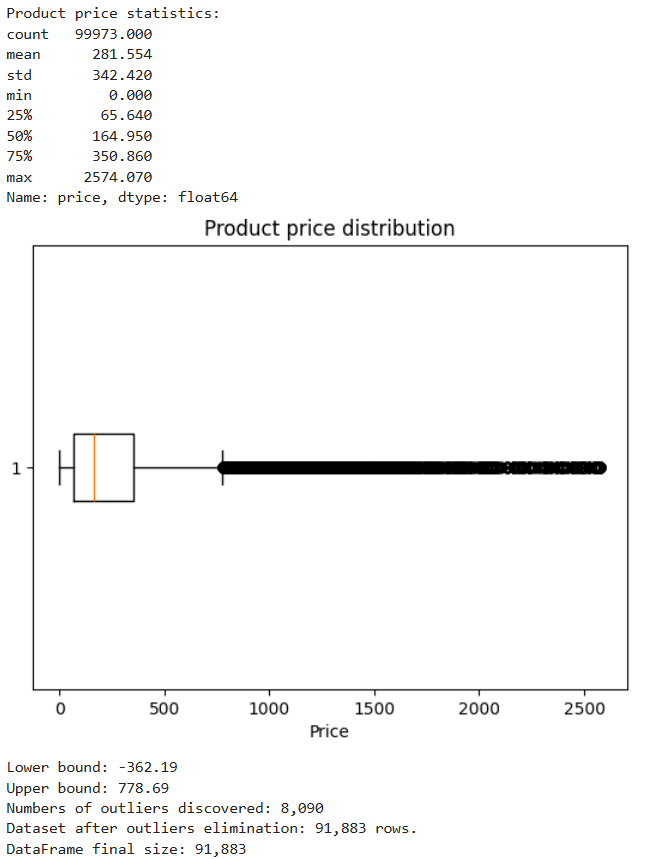
* ***‘hour’***: Giờ trong ngày mà hành vi xảy ra, xác định khung thời gian yêu thích của người dùng (sáng/trưa/tối)
* ***‘day’***: Ngày cụ thể trong tháng, hỗ trợ phát hiện các xu hướng ngắn hạn/khuyến mãi.
* ***'weekday’***: Các ngày trong tuần, phân tích hành vi theo thói quen hàng tuần.
* Giúp mô hình hiểu rõ hơn **bối cảnh thời gian** của hành vi mua hàng, từ đó phát hiện ra **xu hướng tiêu dùng** theo khung giờ/ngày nhất định.

1. **Loại bỏ trùng lặp (Duplicate records)**

Dữ liệu gốc hiện có một số dataset trùng lặp hoàn toàn ở tất cả các cột dữ liệu, có khả năng đến từ lỗi ghi log hay sự cố kĩ thuật trong quá trình thu thập dữ liệu. Nhằm đảm bảo tính tin cậy, nhóm đã **loại bỏ hoàn toàn** các bản ghi trùng lặp, giữ lại 1 bản duy nhất tránh việc mô hình bị học lệch do **lặp lại thông tin nhiều lần**.

1. **Xử lí ngoại lệ (Outliers):**

Khi tiến hành phân tích sơ bộ bằng phương pháp thống kê và boxplot, nhóm nhận thấy ở trường ***‘price’*** có sự phân phối giá không tuân theo phân phối chuẩn, lệch sang phải rõ rệt – nghĩa là phần lớn sản phẩm có giá thấp, trong khi một số ít lại có giá rất cao kéo dài ở phần đuôi bên phải dồ thị.



1. Phát hiện ngoại lai bằng phương pháp Interquartile Range:

Nhóm sử dụng chỉ số khoảng tứ phân vị (IQR) để xác định ranh giới của các giá trị hợp lí. Cụ thể, quá trình thực hiện như sau:

* Tính Q1 và Q3:
* Q1: Giá trị tại phân vị thứ 25%, tức 25% giá trị thấp nhất của dữ liệu nằm dưới Q1
* Q3: Giá trị tại phân vị thứ 75%, tức 75% giá trị thấp nhất của dữ liệu nằm dưới Q3
* Tính IQR: Đây là đại lượng biểu thị phạm vi chứa **phần lớn các giá trị “bình thường”** của dữ liệu.

**IQR = Q3 – Q1**

* Xác định ngưỡng dưới và ngưỡng trên:
* Ngưỡng dưới (Lower Bound): Q1 – 1.5\*IQR
* Ngưỡng trên (Upper Bound): Q3 + 1.5\*IQR
* Xác định outliers: Bất kể giá trị nào **nhỏ hơn ngưỡng dưới** hoặc **lớn hơn ngưỡng trên** được xem là outliers.

1. Loại bỏ hoặc điều chỉnh giá trị ngoại lai:

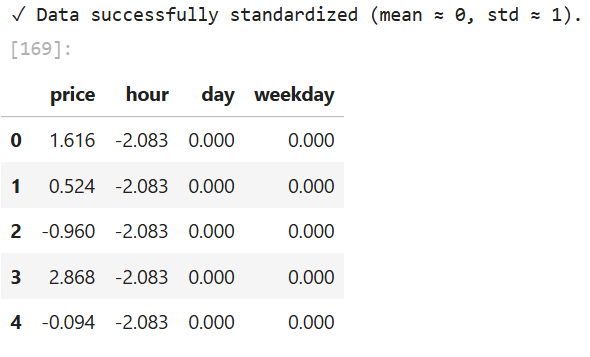
Sau khi phát hiện, nhóm tiến hành xử lý các điểm ngoại lai theo hướng sau:

* Các bản ghi có giá trị lớn hơn ngưỡng trên được loại bỏ khỏi tập dữ liệu huấn luyện. Việc loại bỏ giúp mô hình không ảnh hưởng bởi các giá trị cực đoan.
* Các bản ghi có giá trị thấp bất thường (<=0) cũng được loại bỏ hoàn toàn do không phản ánh tình huống thực tế của sản phẩm thương mại.

1. Kiểm tra lại sau xử lí:

Sau khi tiến hành loại bỏ, nhóm nhận thấy:

* Phân phối giá cân đối hơn, không còn các giá trị cực đoan kéo dài ở phần đuôi phải.
* Các chỉ số trung bình, trung vị, độ lệch chuẩn ổn định hơn.
* Tập dữ liệu phản ánh chính xác hơn xu hướng giá thực tế của sản phẩm, giúp thuật toán hoạt động ổn định và hiệu quả hơn.



1. **Phân loại tệp dữ liệu huấn luyện – kiểm thử:**

Sau khi hoàn tất các bước trên, nhóm tiến hành bước chuẩn bị dữ liệu đầu vào cho mô hình Naïve Bayes. Mục tiêu của giai đoạn này là xây dựng biến mục tiêu, lựa chọn các đặc trưng có ý nghĩa, và chia dữ liệu thành hai phần traning – testing để đánh giá hiệu năng mô hình một cách khách quan.

1. Tạo biến mục tiêu (Target Variable)

Trong bài toán này, nhóm tập trung vào việc xác định người dùng có thực hiện hành vi mua hàng hay không. Do đó, một biến mục tiêu mới được tạo ra là *‘is\_purchase’*, có giá trị nhị phân như sau:

* is\_purchase = 1 nếu cột ‘*event\_type’* có giá trị ‘purchase’, tức người dùng thực sự mua hàng.
* is\_purchase = 0 nếu cột *‘event\_type’* là *‘view’, ‘cart’* hoặc bất kì hành vi nào không phải mua hàng.
* Biến *‘is\_purchase’* có vai trò là nhãn phân lớp mà mô hình cần dự đoán, chuyển bài toán sang phân loại nhị phân.

1. Lựa chọn đặc trưng (Feature Selection)

Nhóm tiến hành chọn lọc những biến đầu vào **quan trọng** và có ý nghĩa với hành vi mua hàng, gồm ***‘category\_code’, ‘brand’, ‘price’, ‘hour’, ‘day’, ‘weekday’***

1. Phân chia dữ liệu (Traning-Testing Split)

Sau khi xác định rõ biến mục tiêu và tập đặc trưng, dữ liệu được chia thành 2 phần:

* 80% dữ liệu dành cho huấn luyện (training set) – dạy mô hình học được mối quan hệ giữa các đặc trưng và nhãn mục tiêu.
* 20% dữ liệu dành cho kiểm thử (testing set) – đánh giá hiệu năng mô hình trên dữ liệu chưa từng thấy trước đó.

Nhóm sử dụng hàm *train\_test\_split()* với tham số *stratify=y*, giúp giữ nguyên tỉ lệ giữa hai lớp trong cả hai tập dữ liệu, tránh học lệch do mất cân bằng lớp.

1. **Mã hóa và chuẩn hóa dữ liệu:**
2. Encoding:

Trong bộ dữ liệu, hai cột *‘category\_code’* và ‘*brand*’ là các biến dạng chuỗi, nhóm sử dụng LabelEncoder() để tiến hành mã hóa, gán cho mỗi giá trị duy nhất một số nguyên tương ứng, giúp mô hình Naïve Bayes đảm bảo chỉ làm việc với dữ liệu dạng số.

1. Chuẩn hóa dữ liệu:

Các cột số như *‘price’, ‘hour’, ‘day’, ‘weekday’* có đơn vị đo lường khác nhau, nếu không tiến hành chuẩn hóa khiến cho các biến có giá trị lớn (như ‘*price*’) chi phối mô hình, làm giảm ảnh hưởng của các biến nhỏ hơn. Để khắc phục, nhóm áp dụng chuẩn hóa theo Z-score bằng StandardScaler, đưa tất cả các biến số về thang đo có trung bình bằng 0 và độ lệch chuẩn bằng 1. Nhờ đó, mô hình có thể xử lý các đặc trưng với quy mô khác nhau một cách cân bằng hơn.

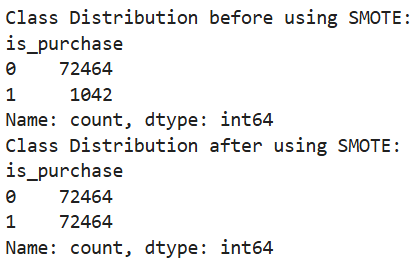
1. Cân bằng dữ liệu bằng kĩ thuật SMOTE:

Qua phân tích ban đầu, biến mục tiêu *‘is\_purchase’* cho thấy sự mất cân bằng nghiêm trọng giữa hai lớp:

* Lớp 0 (không mua hàng) chiếm tỉ lệ rất lớn.
* Lớp 1 (mua hàng) chiến tỉ lệ rất nhỏ.

Nếu để nguyên, mô hình sẽ có xu hướng dự đoán tất cả đều không mua làm giảm độ hiệu quả. Nhóm đã tiến hành sử dụng SMOTE (Synthetic Minority Over-sampling Technique) nhằm tăng cường số lượng mẫu của lớp thiể số bằng cách tạo ra các điểm dữ liệu tổng hợp (không sao chép y nguyên). Kĩ thuật này hoạt động bằng cách chọn ngẫu nhiên một điểm dữ liệu của lớp thiểu số, sau đó tạo thêm các mẫu mới ở giữa các điểm đó và các điểm lân cận (K-Nearest Neighbours), giúp mở rộng không gian dữ liệu một cách tự nhiên.

Quan trọng hơn, SMOTE chỉ áp dụng trên *X\_train, y\_train* để tránh rò rỉ thông tin sang tập kiểm thử.

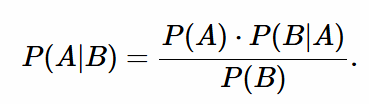


* Số lượng mẫu được cân bằng, giúp mô hình học các đặc trưng của lớp “mua” tốt hơn, cải thiện recall và khả năng phát hiện đúng hành vi mua hàng thực tế.

1. **Huấn luyện mô hình Gaussian Naïve Bayes:**

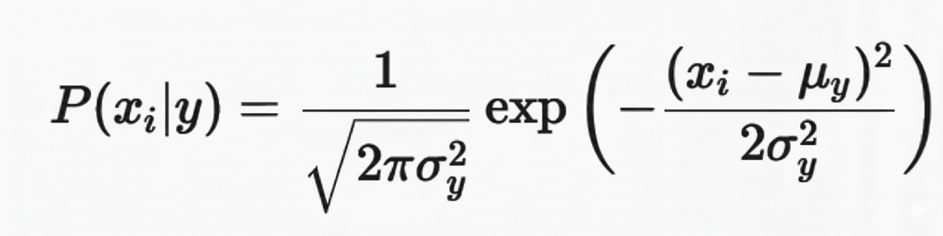
Bước tiếp theo của quy trình là xây dựng và huấn luyện mô hình Gaussian Naïve Bayes – một trong các biến thể phổ biến nhất của thuật toán Naïve Bayes.

Nguyên lý hoạt động: Thuật toán Naïve Bayes dựa trên định lý xác suất Bayes, mô tả mối quan hệ giữa xác suất của nhãn (lớp) và xác suất có điều kiện của các đặc trưng (features) quan sát được:



Naïve Bayes giả định các đặc trưng đầu vào độc lập nhau khi biết lớp nhãn. Điều này giúp việc tính toán trở nên rất nhanh chóng, vì mô hình chỉ cần ước lượng phân phối xác suất của từng đặc trưng riêng biệt thay vì toàn bộ tập hợp đặc trưng cùng lúc.

Biến thể Gaussian Naïve Bayes: Sau khi chuẩn hóa, các đặc trưng có thể được coi như tuân theo phân phối chuẩn Gauss. Với mỗi lớp (0/1), mô hình ước lượng trung bình và độ lệch chuẩn của từng đặc trưng đã được xác định, ta có thể áp dụng công thức hàm mật độ xác suất của phân phối chuẩn:

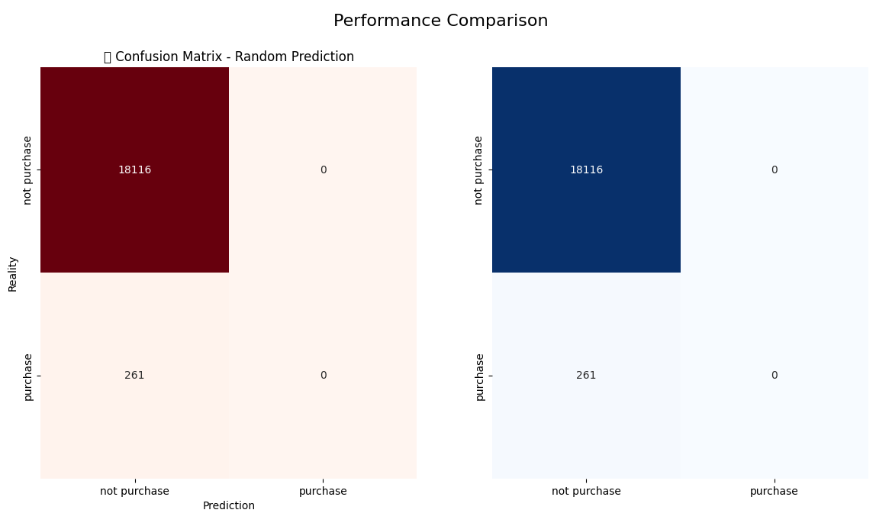


Quy trình huấn luyện:

* Sử dụng tập dữ liệu đã được cân bằng bằng SMOTE (*X\_train\_smote, y\_train\_smote*)
* Khởi tạo mô hình Gaussian Naïve Bayes bằng thư viện Scikit-learn
* Huấn luyện mô hình với tập huấn luyện model.fit
* Sau khi huấn luyện, mô hình học được phân phối xác suất của từng đặc trưng trong từng lớp, từ đó có thể dự đoán xác suất người dùng mua hàng khi biết các yếu tố đầu vào.

1. **So sánh hiệu quả mô hình:**

Để đánh giá hiệu quả thực tế, **mô hình Naive Bayes** được so sánh trực tiếp với một **mô hình cơ sở ngẫu nhiên (Random Baseline)**. Mô hình ngẫu nhiên này được thiết lập để dự đoán "mua hàng" với cùng tần suất mà hành vi "mua hàng" thực sự xảy ra trong tập dữ liệu, tạo ra một thước đo hiệu suất tối thiểu cần phải vượt qua.



Thông qua hai ma trận nhầm lẫn này, ta có thể thấy:

* Naïve Bayes có số lượng True Positives **cao hơn** một cách rõ rệt, điều này khẳng định mô hình **đã học được** các mẫu hành vi liên quan đến quyết định mua hàng.
* Mô hình ngẫy nhiên tạo ra số lượng False Positives **lớn**, cho thấy các dự đoán của nó thiếu cơ sở và độ tin cậy không cao.

1. **Kết luận:**

Dự án đã **xây dựng thành công mô hình Naive Bayes** để dự đoán hành vi mua hàng của khách hàng, chứng minh rằng mô hình có khả năng học được các quy luật cơ bản từ dữ liệu và cho kết quả **vượt trội hơn hẳn so với dự đoán ngẫu nhiên**. Mặc dù đã đạt mục tiêu ban đầu, mô hình vẫn còn một số hạn chế, đặc biệt là về chỉ số Precision.

Hướng cải tiến:

* Thử nghiệm các thuật toán phức tạp hơn: Triển khai các mô hình như **Logistic Regression**, **Random Forest**, hoặc **Gradient Boosting** để so sánh và tìm ra thuật toán tối ưu nhất cho bài toán này.
* Kỹ thuật đặc trưng (Feature Engineering): Xây dựng thêm các thuộc tính mới từ dữ liệu hiện có (ví dụ: tần suất truy cập, thời gian trung bình trên trang) để cung cấp cho mô hình nhiều thông tin hữu ích hơn.
* Tinh chỉnh tham số (Hyperparameter Tuning): Áp dụng các kỹ thuật như Grid Search để tìm ra bộ tham số tối ưu, giúp nâng cao độ chính xác của mô hình.