**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A blue shield with white text and red flames

Description automatically generated**

**TIỂU LUẬN MÔN HỌC**

**KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG**

**ỨNG DỤNG THUẬT TOÁN PHÂN LOẠI**

**VÀO BÀI TOÁN PHÂN LOẠI TIẾP THỊ NGÂN HÀNG**

**Giảng viên hướng dẫn: VÕ THỊ HỒNG THẮM**

**Sinh viên thực hiện: PHAN QUỐC ĐIỀN**

**MSSV: 2100007729**

**Môn học: Khai thác dữ liệu và ứng dụng**

**Lớp: 21DTH2B**

**Khoá: 2021**

**Tp.HCM, tháng 01 năm 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A blue shield with white text and red flames

Description automatically generated**

**TIỂU LUẬN MÔN HỌC**

**KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG**

**ỨNG DỤNG THUẬT TOÁN PHÂN LOẠI**

**VÀO BÀI TOÁN PHÂN LOẠI TIẾP THỊ NGÂN HÀNG**

**Giảng viên hướng dẫn: VÕ THỊ HỒNG THẮM**

**Sinh viên thực hiện: PHAN QUỐC ĐIỀN**

**MSSV: 2100007729**

**Môn học: Khai thác dữ liệu và ứng dụng**

**Lớp: 21DTH2B**

**Khoá: 2021**

**Tp.HCM, tháng 01 năm 2024**

BM-ChT

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH KÌ THI KẾT THÚC HỌC PHẦN**

**TRUNG TÂM KHẢO THÍ HỌC KÌ I NĂM HỌC 2023 - 2024**

🙜 🙜 🙝 🙝 🙜 🙜 🙝 🙝

**PHIẾU CHẤM THI TIỂU LUẬN**

Môn thi: **Khai thác dữ liệu và ứng dụng** Mã lớp học phần: **21DTH2B**

Nhóm sinh viên thực hiện:

3. **Phan Quốc Điền** Tham gia đóng góp: **100%**

Ngày thi: **29/01/2024** Phòng thi: **L.708**

Đề tài tiểu luận/đồ án của sinh viên: **Ứng dụng các thuật toán phân loại vào bài toán phân loại tiếp thị ngân hàng**

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí (theo CĐR HP)** | **Đánh giá của giáo viên** | **Điểm tối đa** | **Điểm đạt được** |
| Cấu trúc của báo cáo |  |  |  |
| Nội dung |  |  |  |
| - Các nội dung thành phần |  |  |  |
| - Lập luận |  |  |  |
| - Kết luận |  |  |  |
| Trình bày |  |  |  |
| **TỔNG ĐIỂM** |  | 10 |  |

**Giảng viên chấm thi**

(*Ký và ghi rõ họ tên*)

TS. Võ Thị Hồng Thắm

# CHƯƠNG 1: GIỚI THIỆU

## 1.1. Giới thiệu đề tài

Trong thời kỳ công nghiệp 4.0, các phương pháp thông minh nhân tạo đang ngày càng trở thành một công cụ hữu ích trong nhiều lĩnh vực, đặc biệt là trong lĩnh vực ngân hàng. Trong ngữ cảnh này, đề tài này tập trung vào việc áp dụng thuật toán phân loại để phân loại tiếp thị 1 ngân hàng ở Bồ Đào Nha.

## 1.2. Lý do chọn đề tài

Vấn đề phân loại tiếp thị ngân hàng là một thách thức quan trọng. Khi hiểu rõ về các yếu tố quyết định giá trị sẽ giúp các bên liên quan, nhân viên tiếp thị và đưa ra quyết định nhanh chóng, thông minh.

## 1.3. Mục tiêu của đề tài

Mục tiêu của nghiên cứu này là phát triển một mô hình để phân loại đối tượng tiếp thị. Bằng cách này, chúng ta có thể đưa ra phân loại đối tượng tiếp thị dựa trên các đặc điểm quan trọng được đề cập trong tập dữ liệu.

## 1.4. Phương pháp đề tài

Phương pháp nghiên cứu sẽ bao gồm các bước chính: thu thập/tải dữ liệu, tiền xử lý dữ liệu, xây dựng mô hình Logistic Regression, Decision Tree, Random Forest và đánh giá hiệu suất của mô hình thông qua các chỉ số như Accuracy, F1-Score, Precision-Score.

## 1.5. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài là thị trường bất động sản ở Bồ Đào Nha, với những với những đối tượng tiếp thị đa dạng. Phạm vi nghiên cứu bao gồm việc phát triển một mô phân loại đối tượng tiếp thị có khả năng áp dụng rộng rãi và đáp ứng được nhu cầu của ngân hàng.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. Thuật toán phân loại

### 2.1.1. Tổng quát về Logistic

Hồi quy logistic đóng một vai trò quan trọng trong lĩnh vực trí tuệ nhân tạo và máy học (AI/ML). Mô hình máy học là những chương trình phần mềm có khả năng tự đào tạo để thực hiện các nhiệm vụ xử lý dữ liệu phức tạp mà không yêu cầu sự can thiệp của con người. Hồi quy logistic trong các mô hình máy học có thể giúp tổ chức rút ra thông tin sâu rộng từ dữ liệu kinh doanh của họ. Những thông tin này có thể được sử dụng để phân tích và dự đoán, từ đó giảm chi phí, tăng hiệu suất và thích nghi với quy mô nhanh chóng. Dưới đây là một số lợi ích cụ thể của việc sử dụng hồi quy logistic so với các kỹ thuật máy học khác:

* **Tính Đơn Giản:**

Các mô hình hồi quy logistic ít phức tạp toán học hơn so với một số phương pháp máy học khác. Điều này làm cho chúng dễ triển khai ngay cả khi đội ngũ không có chuyên môn sâu về máy học.

* **Tốc Độ:**

Hồi quy logistic có khả năng xử lý dữ liệu lớn ở tốc độ cao vì chúng yêu cầu ít tài nguyên tính toán hơn, như bộ nhớ và sức mạnh xử lý. Điều này làm cho chúng trở thành lựa chọn lý tưởng cho các tổ chức mới bắt đầu với các dự án máy học để đạt được kết quả nhanh chóng.

* **Sự Linh Hoạt:**

Hồi quy logistic có thể được sử dụng để giải quyết các vấn đề phân loại có hai hoặc nhiều kết quả hữu hạn. Chúng cũng có thể được áp dụng để tiền xử lý dữ liệu, chẳng hạn như giảm độ lớn của phạm vi giá trị, tạo điều kiện thuận lợi cho các kỹ thuật máy học khác.

* **Khả Năng Hiển Thị:**

Phân tích hồi quy logistic mang lại khả năng nhìn nhận rõ ràng về các quy trình phần mềm nội bộ, giúp nhà phát triển dễ dàng xác định và sửa lỗi. Đồng thời, tính đơn giản của chúng giúp hiệu suất và sửa lỗi trở nên dễ dàng hơn.

### 2.1.2. Tổng quát về Decision Tree

Cây quyết định (Decision Tree) là một mô hình máy học rất mạnh mẽ và linh hoạt được sử dụng cho cả các tác vụ phân loại và dự đoán. Nó có thể được áp dụng vào nhiều lĩnh vực, bao gồm kinh doanh, y tế, tài chính, và nhiều lĩnh vực khác. Dưới đây là một tổng quan về cây quyết định:

Nguyên tắc hoạt động:

Cây quyết định hoạt động bằng cách xây dựng một cây có các quy tắc quyết định dựa trên dữ liệu đào tạo. Mỗi nút trong cây đại diện cho một quyết định dựa trên một thuộc tính cụ thể, và các nhánh từ nút biểu thị các khả năng kết quả khác nhau dựa trên giá trị của thuộc tính đó.

* **Thuộc tính chọn lọc:**

Khi xây dựng cây, thuật toán chọn thuộc tính tại mỗi nút sao cho nó giúp tối ưu hóa quyết định. Một số tiêu chí phổ biến bao gồm Độ thuần túy Gini (Gini impurity) cho bài toán phân loại và Sự giảm độ hồi quy (Reduction in Variance) cho bài toán dự đoán.

* **Quyết định đệ quy:**

Quá trình xây dựng cây là một quá trình đệ quy, mỗi lần chia nhánh tạo ra các nút con và tiếp tục đến khi một điều kiện dừng được đạt đến, ví dụ như độ sâu tối đa của cây, số lượng mẫu tối thiểu tại mỗi lá, hoặc không còn khả năng chia nhánh nào mang lại lợi ích.

* **Quyết định rõ ràng và giải thích:**

Mô hình cây quyết định tạo ra các quy tắc quyết định rõ ràng và dễ giải thích. Các quy tắc này có thể được hiểu và kiểm tra bởi các chuyên gia và người không chuyên.

* **Khả năng xử lý cả dữ liệu không đồng nhất và dữ liệu thiếu:**

Cây quyết định có khả năng xử lý cả dữ liệu không đồng nhất và dữ liệu thiếu một cách linh hoạt, làm cho chúng trở thành một lựa chọn hữu ích cho nhiều tình huống thực tế.

* **Nguy cơ quá mức khớp (Overfitting):**

Mặc dù cây quyết định có thể tạo ra quy tắc rõ ràng, nhưng nếu không được kiểm soát, chúng có thể dễ bị quá mức khớp với dữ liệu đào tạo, tức là quá mức thích ứng với nhiễu và không tổng quát hóa tốt cho dữ liệu mới.

* **Phương tiện học (Ensemble Learning):**

Cây quyết định thường được sử dụng như một thành phần trong các kỹ thuật học phối hợp như Random Forest và Gradient Boosting, để cải thiện tính tổng quát và khả năng dự đoán.

### 2.1.3. Tổng quát về Random Forest

Random Forest là một phương pháp máy học thuộc loại Ensemble Learning, được xây dựng trên cơ sở việc kết hợp nhiều cây quyết định để tạo ra một mô hình mạnh mẽ và ổn định. Dưới đây là một tổng quan về Random Forest:

* **Nguyên tắc hoạt động:**

Random Forest sử dụng kỹ thuật học phối hợp để tạo ra một tập hợp các cây quyết định (Decision Trees). Mỗi cây được đào tạo trên một phần ngẫu nhiên của dữ liệu đào tạo và sử dụng một phần ngẫu nhiên của các thuộc tính trong quá trình xây dựng cây.

* **Bootstrap Aggregating (Bagging):**

Random Forest sử dụng kỹ thuật Bagging để tạo ra các tập dữ liệu con (bootstrap samples) bằng cách lấy mẫu ngẫu nhiên từ tập dữ liệu đào tạo với sự thay thế. Mỗi cây được đào tạo trên một tập dữ liệu con riêng biệt.

* **Ngẫu nhiên chọn thuộc tính:**

Trong quá trình xây dựng mỗi cây quyết định, Random Forest ngẫu nhiên chọn một số lượng nhỏ thuộc tính để xem xét tại mỗi nút quyết định. Điều này giúp làm giảm nguy cơ quá mức khớp và tăng tính tổng quát của mô hình.

* **Voting hoặc Average:**

Khi có một tập hợp các cây quyết định được xây dựng, Random Forest thực hiện quyết định cuối cùng dựa trên phương thức voting (đối với bài toán phân loại) hoặc trung bình (đối với bài toán hồi quy) các dự đoán của từng cây.

* **Khả năng xử lý nhiễu và overfitting:**

Random Forest thường có khả năng xử lý tốt việc nhiễu và quá mức khớp. Bởi vì mỗi cây được xây dựng trên một tập dữ liệu con ngẫu nhiên, và chỉ một phần nhỏ của các thuộc tính được sử dụng ở mỗi nút, mô hình trở nên ít nhạy cảm hơn với dữ liệu đào tạo cụ thể.

* **Ứng dụng rộng rãi:**

Random Forest được sử dụng rộng rãi trong nhiều lĩnh vực như phân loại ảnh, dự đoán giá cổ phiếu, chẩn đoán y tế, và nhiều ứng dụng khác do tính linh hoạt và hiệu suất cao của nó.

* **Tăng tốc độ tính toán:**

Mặc dù Random Forest có thể yêu cầu nhiều tài nguyên tính toán khi xây dựng nhiều cây, nhưng quá trình này có thể được tăng tốc bằng cách sử dụng đa luồng (multithreading) hoặc xử lý song song.

### 2.1.4. Tổng quát về KNN

K-Nearest Neighbors (KNN) là một thuật toán máy học giám sát được sử dụng cho cả các tác vụ phân loại và dự đoán. Nó hoạt động dựa trên nguyên lý rằng các mẫu cùng lớp thường có đặc điểm tương tự và nằm gần nhau trong không gian đặc trưng. Dưới đây là một tổng quan về KNN:

* **Nguyên tắc hoạt động:**

KNN hoạt động dựa trên nguyên tắc đơn giản: "Những người láng giềng gần nhất có xu hướng giống nhau." Đối với một điểm dữ liệu mới, KNN tìm k mẫu gần nhất trong không gian đặc trưng và quyết định lớp của điểm dữ liệu mới dựa trên đa số lớp của những mẫu gần nhất.

* **Tham số K:**

K là số láng giềng gần nhất mà thuật toán sẽ xem xét để đưa ra quyết định. Lựa chọn giá trị K quan trọng, và nó có thể ảnh hưởng đến hiệu suất của mô hình. Một K nhỏ có thể làm cho mô hình nhạy cảm với nhiễu, trong khi K lớn có thể làm giảm độ chính xác và độ nhạy.

* **Hàm đo khoảng cách:**

Để xác định "gần nhất," KNN sử dụng hàm đo khoảng cách như Euclidean distance, Manhattan distance, hoặc Minkowski distance. Hàm đo khoảng cách này quyết định cách tính khoảng cách giữa các điểm dữ liệu trong không gian đặc trưng.

* **Không gian đặc trưng:**

KNN yêu cầu một không gian đặc trưng trong đó các điểm dữ liệu có thể được đo khoảng cách. Việc chuẩn bị dữ liệu và lựa chọn các đặc trưng quan trọng đóng vai trò quan trọng trong hiệu suất của thuật toán.

* **Khả năng làm việc với dữ liệu không cân bằng:**

KNN có thể làm việc tốt với các tập dữ liệu không cân bằng, nơi có một lớp chiếm số lượng lớn hơn so với các lớp khác.

* **Khả năng dự đoán và phân loại:**

KNN có thể được sử dụng cho cả bài toán dự đoán (regression) và phân loại (classification), tùy thuộc vào việc quyết định giá trị trung bình của các láng giềng (đối với dự đoán) hoặc đa số lớp của chúng (đối với phân loại).

* **Tính đơn giản và linh hoạt:**

Mặc dù có nhược điểm, KNN thường được xem là một thuật toán đơn giản và linh hoạt, đặc biệt là đối với các vấn đề có số lượng mẫu nhỏ và không có giả định về phân phối của dữ liệu.

### 2.1.5. Tổng quát về GridSearchCV

GridSearchCV (Grid Search Cross-Validation) là một kỹ thuật tìm kiếm siêu tham số phổ biến trong máy học. Nó giúp tự động thử nghiệm và lựa chọn các giá trị tối ưu cho các tham số của một mô hình thông qua việc duyệt qua một lưới các giá trị thử nghiệm được xác định trước đó. Dưới đây là một tổng quan về GridSearchCV:

* **Tìm kiếm trên lưới:**

GridSearchCV thực hiện tìm kiếm trên lưới các giá trị thử nghiệm cho các tham số của mô hình. Các giá trị này được xác định trước đó và tạo thành một lưới các kết hợp thử nghiệm.

* **Cross-Validation:**

Để đánh giá hiệu suất của mỗi kết hợp thử nghiệm, GridSearchCV thường sử dụng phương pháp Cross-Validation. Nó chia tập dữ liệu thành các tập huấn luyện và tập kiểm tra, và sau đó đánh giá mô hình trên tập kiểm tra. Quá trình này được lặp lại với các phần khác nhau của dữ liệu để đảm bảo độ tin cậy của kết quả.

* **Tham số Mô hình:**

GridSearchCV có thể áp dụng cho bất kỳ mô hình nào có các tham số cần được điều chỉnh. Điều này có thể bao gồm các mô hình như Support Vector Machines, Random Forest, K-Nearest Neighbors, và nhiều mô hình khác.

* **Hiệu suất:**

Hiệu suất của mỗi kết hợp thử nghiệm được đánh giá thông qua một phép đo hiệu suất được chọn trước, chẳng hạn như độ chính xác (accuracy), độ F1 (F1 score), hoặc RMSE (Root Mean Squared Error) tùy thuộc vào bài toán cụ thể (phân loại hoặc dự đoán).

* **Tối ưu hóa tham số:**

GridSearchCV tìm kiếm và chọn bộ tham số tối ưu cho mô hình dựa trên hiệu suất trung bình qua các lượt Cross-Validation. Bộ tham số tối ưu là bộ tham số mà làm cho hiệu suất trên tập kiểm tra là tốt nhất.

* **Tài nguyên tính toán:**

Do GridSearchCV thử nghiệm mọi kết hợp có thể của các giá trị tham số, nó có thể yêu cầu nhiều tài nguyên tính toán. Điều này có thể là một hạn chế đối với các tập dữ liệu lớn hoặc khi số lượng tham số cần tối ưu là lớn.

* **Tự động hóa:**

GridSearchCV giúp tự động hóa quá trình tìm kiếm siêu tham số, giúp giảm công sức và thời gian so với việc thử nghiệm thủ công từng giá trị tham số.

# CHƯƠNG 3: XÂY DỰNG MÔ HÌNH

## 3.1. Xây dựng mô hình với thư viện scikit-learn của Python

Scikit-learn (sklearn) là một thư viện mã nguồn mở phổ biến trong ngôn ngữ lập trình Python được sử dụng cho việc phân tích và xử lý dữ liệu, cũng như xây dựng và huấn luyện các mô hình máy học. Thư viện này cung cấp nhiều công cụ cho nhiều thuật toán khác nhau, bao gồm cả Losgistic Regression.

Dưới đây là một tổng quan về cách sử dụng sklearn để huấn luyện mô hình Losgistic Regression:



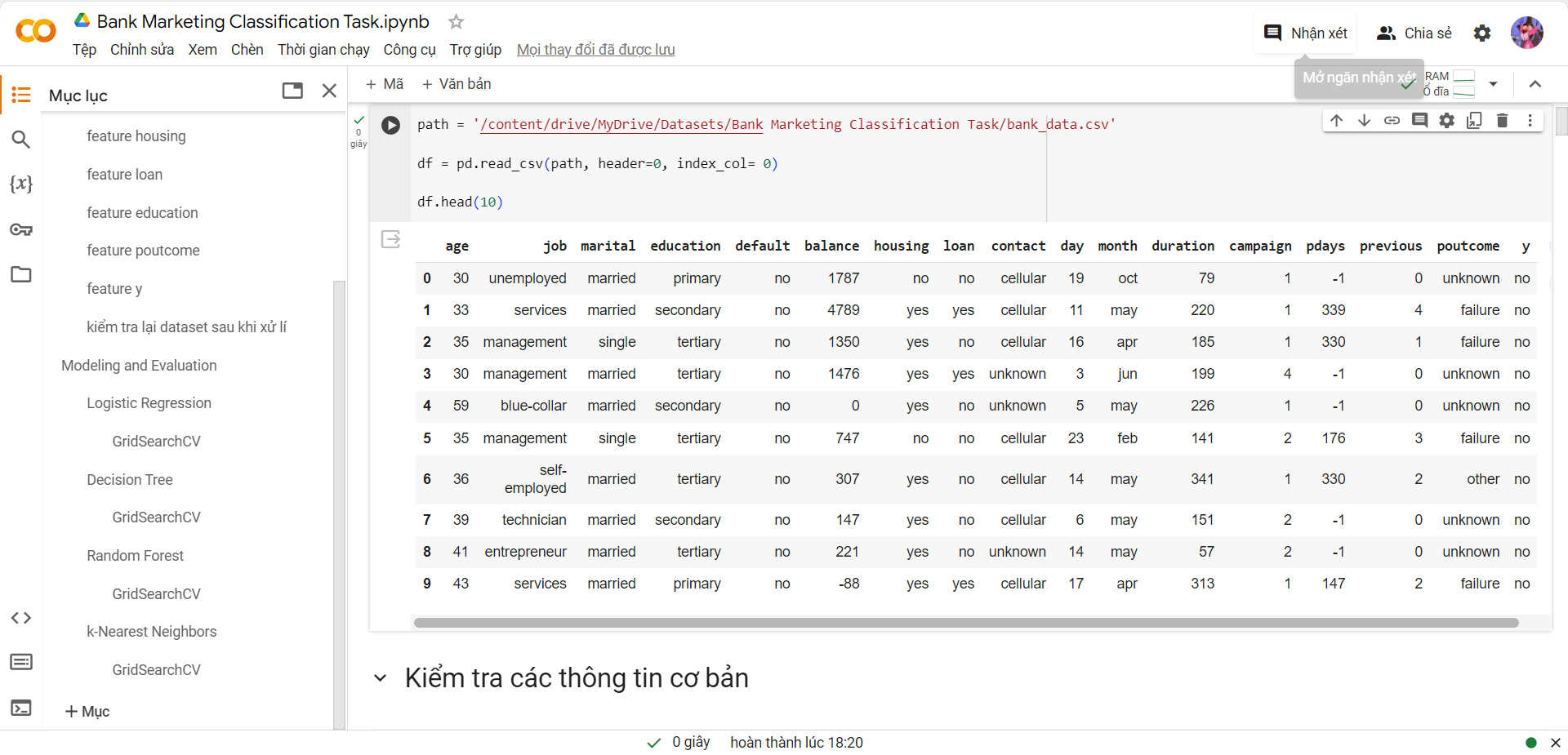
Hình 1: Xây dựng mô hình với thư viện sklearn trong Python

Các thuật toán khác được xây dựng 1 cách tương tự thông qua thư viện sklearn.

# CHƯƠNG 4: THỰC NGHIỆM

## 4.1. Xây dựng bộ dữ liệu

### 4.1.1. Mô tả dữ liệu

Dữ liệu có liên quan đến các chiến dịch tiếp thị trực tiếp của một tổ chức ngân hàng Bồ Đào Nha. Các chiến dịch tiếp thị đều dựa trên các cuộc gọi điện thoại. Thông thường, cần có nhiều liên hệ với cùng một khách hàng để đánh giá xem sản phẩm (tiền gửi có kỳ hạn ngân hàng) có được đăng ký hay không ('yes') hay không ('no'). 

Hình 2: Hình ảnh bộ dữ liệu tiếp thị ngân hàng của 1 ngân hàng ở Bồ Đào Nha

Bộ dữ liệu này gồm có 17 cột được mô tả như sau:

* **age**: Tuổi khách hàng.
* **job**: Công việc của hàng hàng ("admin.", "unknown", "unemployed", "management", "housemaid", "entrepreneur", "student","blue-collar", "self-employed", "retired","technician", "services").
* **marital**: Tình trạng hôn nhân (categorical: "married", "divorced", "single"; note: "divorced" có nghĩa là đã ly hôn hoặc ở goá).
* **education:**( categorical:"unknown", "secondary", "primary", "tertiary")
* **default:** Có tín dụng mặc định không? (binary: "yes","no")
* **balance:** Số dư trung bình hàng năm, tính bằng euro (số)
* **housing:** Có khoản vay mua nhà không? (binary: "yes","no")
* **loan:** Có khoản vay cá nhân hay không? **(binary: "yes","no")**

**Liên quan đến lần liên hệ cuối cùng của chiến dịch hiện tại:**

* **contact:** Loại liên lạc (categorical: "unknown","telephone","cellular")
* **day:** Ngày liên hệ cuối cùng trong tháng (số)
* **month:** Tháng liên hệ cuối cùng (categorical: "jan", "feb", "mar", …, "nov", "dec")
* **duration:** Thời lượng liên lạc lần cuối, tính bằng giây (số)
* **Các thuộc tính khác:**
* **campaign:** Số lượng liên hệ được thực hiện trong chiến dịch này và cho khách hàng này (số, bao gồm liên hệ cuối cùng)
* **pdays:** Số ngày đã trôi qua sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước đó (số, -1 có nghĩa là khách hàng chưa được liên hệ trước đó)
* **previous:** Số lượng liên hệ được thực hiện trước chiến dịch này và cho khách hàng này (số)
* **poutcome:** Kết quả của chiến dịch tiếp thị trước đó (categorical: "unknown","other","failure","success")

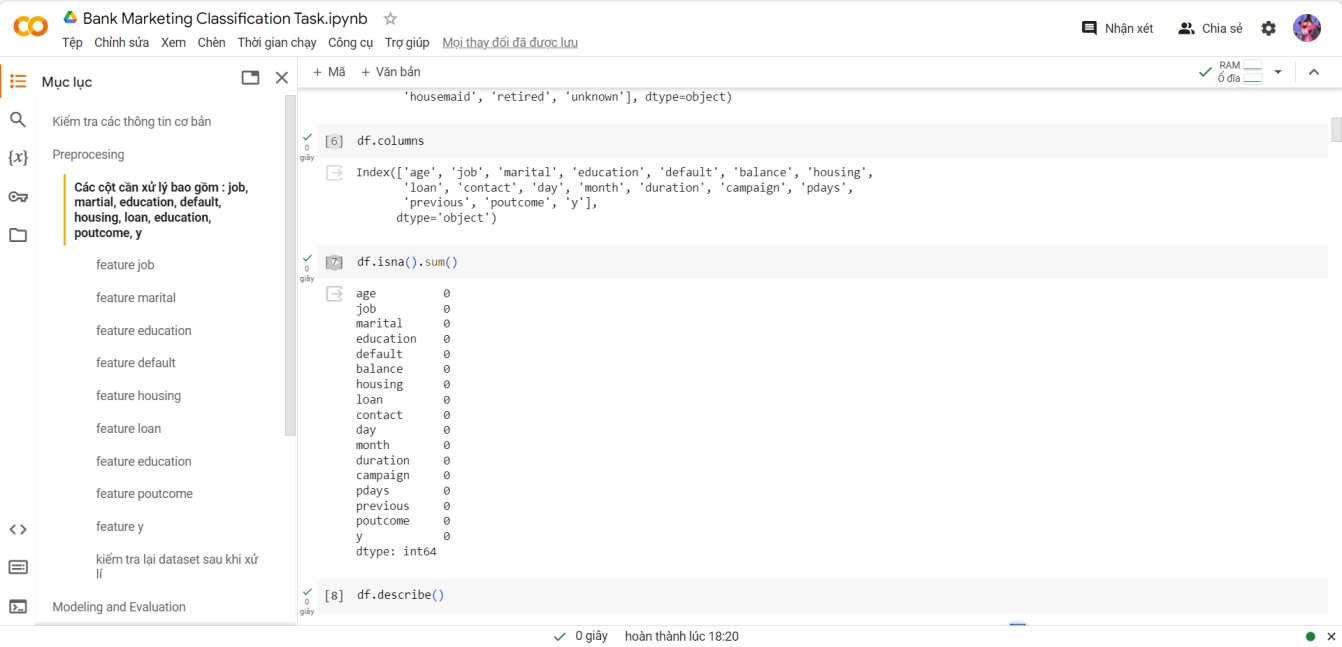
**Biến đầu ra (Kết quả mong muốn):**

* **y:** Khách hàng có đăng ký gửi tiền có kỳ hạn không? (binary: "yes","no")

### 4.1.2. Tiền xử lý dữ liệu

Vì dữ liệu này được tác giả thu thập về vẫn còn thô nên chúng ta cần xử lý lại bộ dữ liệu để nó được sạch sẽ và hoàn thiện hơn để có thể tạo ra một mô hình huấn luyện cho hiệu suất tốt. Quá trình tiền xử lý dữ liệu của tôi bao gồm 9 bước sau:

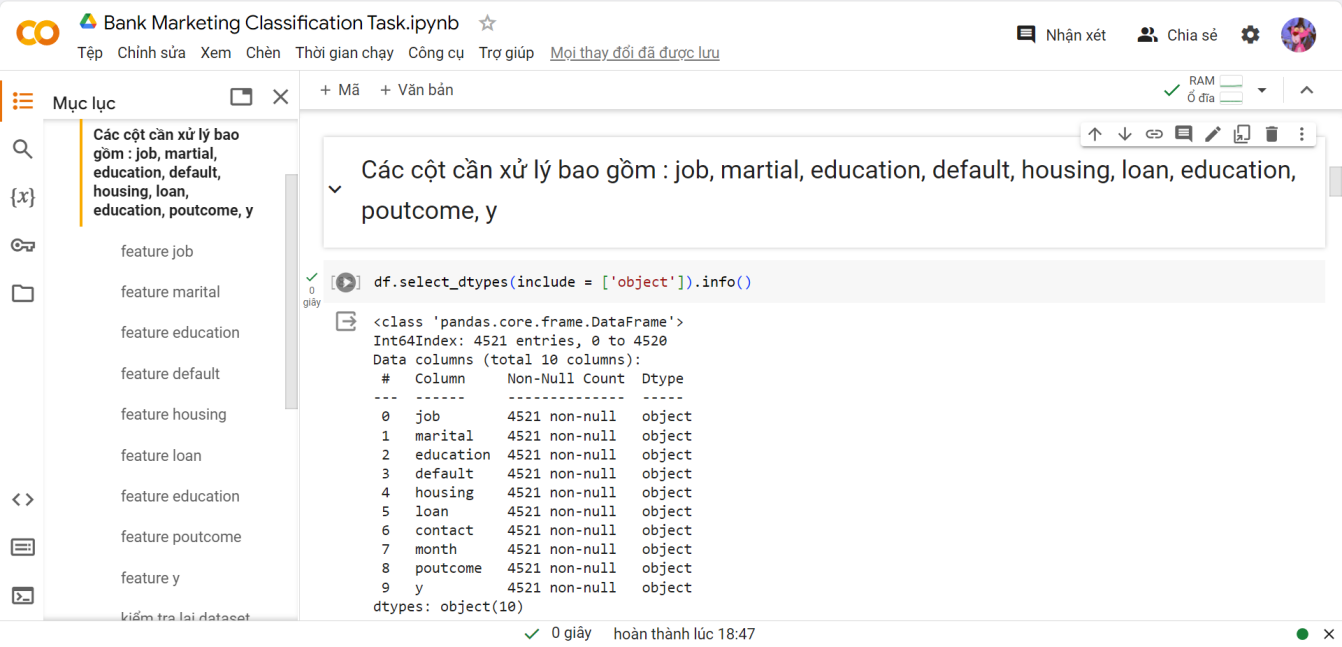
**Bước 1:** Chọn ra những đặc trưng quan trọng

Tập dữ liệu được cung cấp từ ngân hàng đã được chọn những đặc trưng được cho là sẽ có ảnh hưởng tới bài toán phân loại. Vì vậy, tiểu luận này sẽ không đề cập đến các bước lấp đầy dữ liệu bị thiếu và chọn đặc trưng quan trọng. 

Hình 3: Bộ dữ liệu đã được nhà cung cấp chọn lọc kỹ càng trước khi tải lên Kaggle

**Bước 2:** Xử lý feature job

Các cột job, martial, education, default, housing, loan, education, poutcome, y có datatype là object nên cần phải được xử lý trước khi đưa vào phân loại.



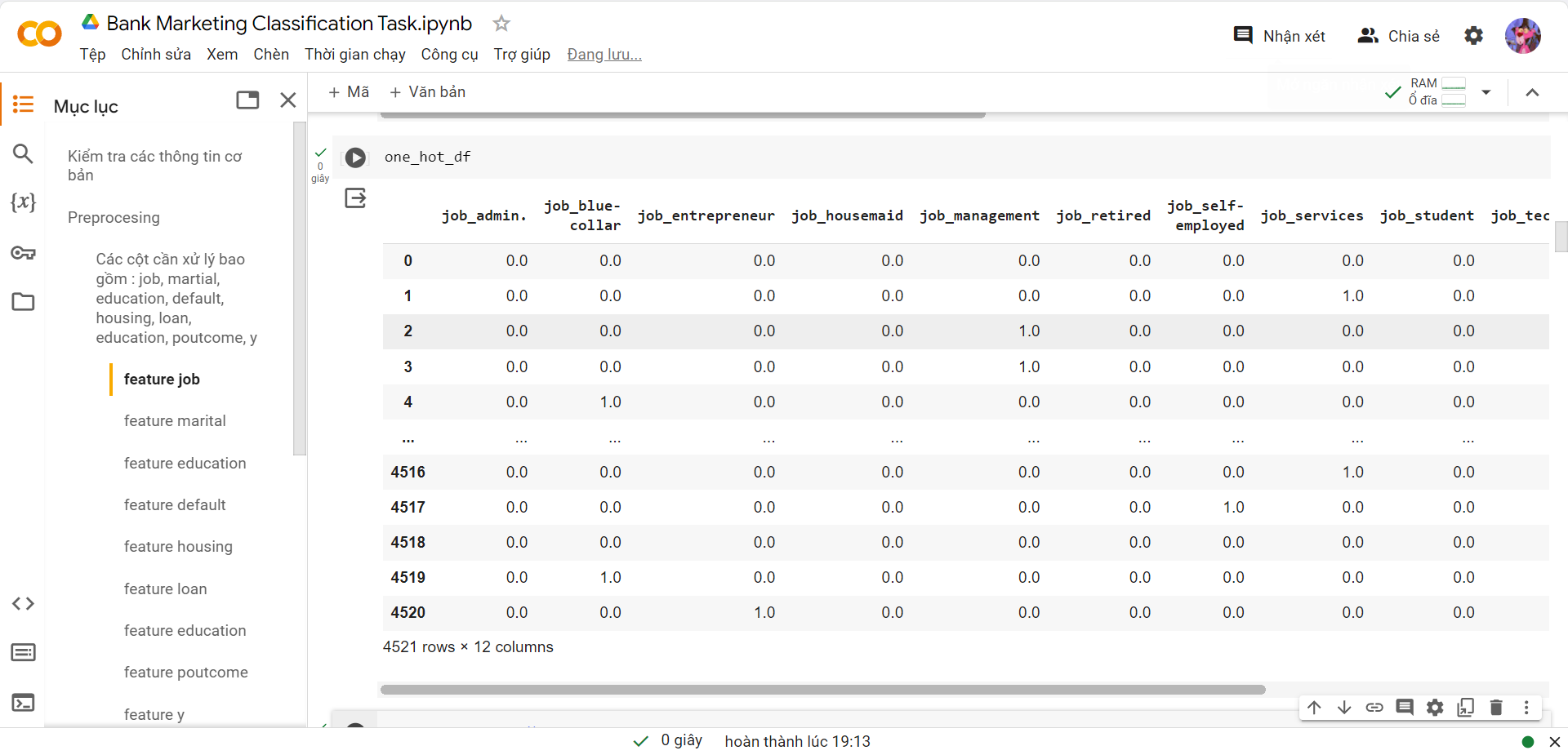
Hình 4: các feature có dtype là object trong dataset

Dựa vào Hình 4 trên, ta thấy rằng giá trị ở các cột chưa phải là các giá trị định tính (biến phân loại). Đồ án này sẽ sử dụng One-hot Encoding và Label Encoding để chuyển đổi các feature này.

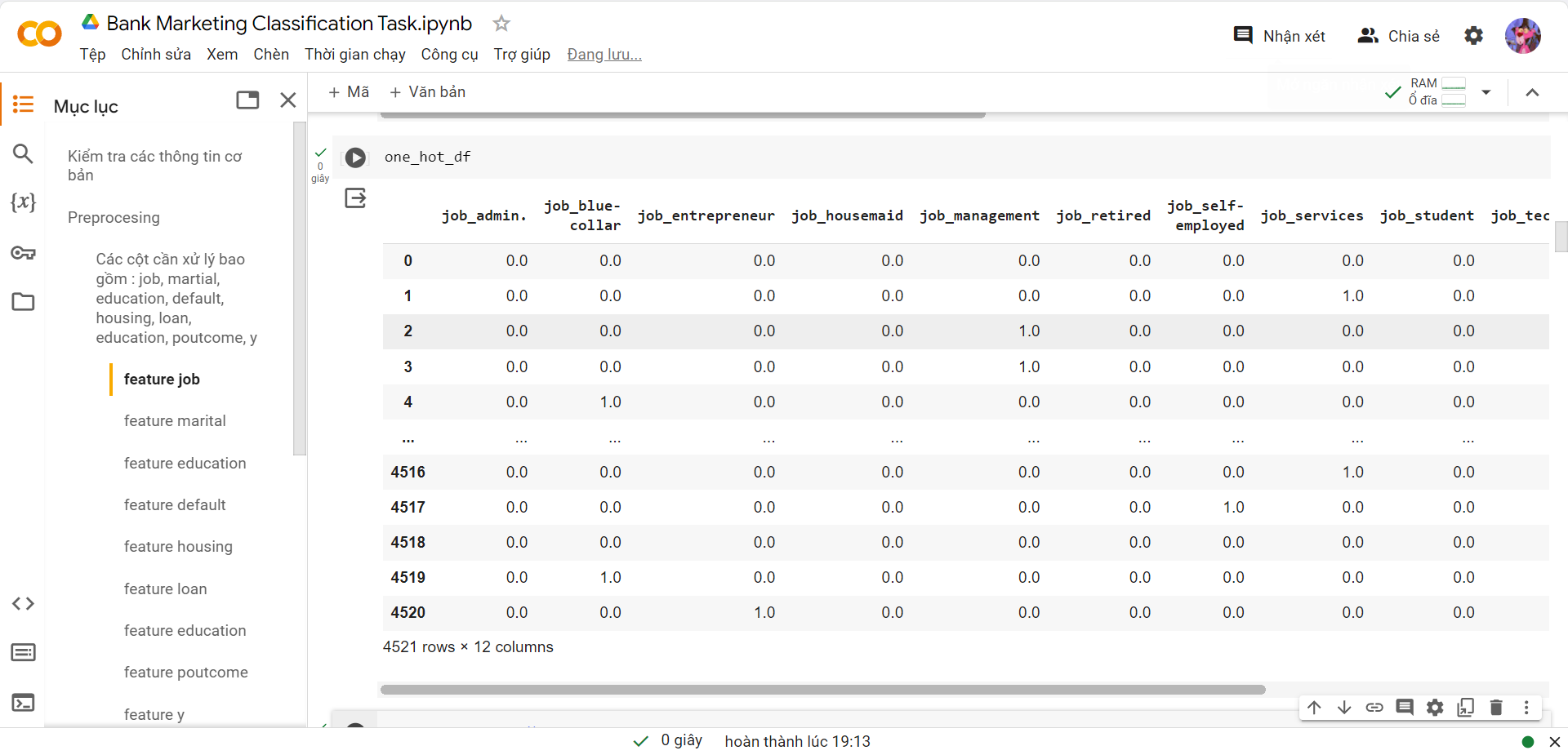
Đầu tiên, import OneHotEncoder và gán cho 1 biến tạm ‘encoder’. Dùng hàm fit\_transform() để chuyển đổi các biến phân loại thành ma trận nhị phân



Hình 5: Hình ảnh xử lý cột job



Hình 6: Hình ảnh các đặc trưng mới sau khi xử lý feature job



Hình 7: Tiến hành gộp các feature mới vào dataset và xoá đi feature job

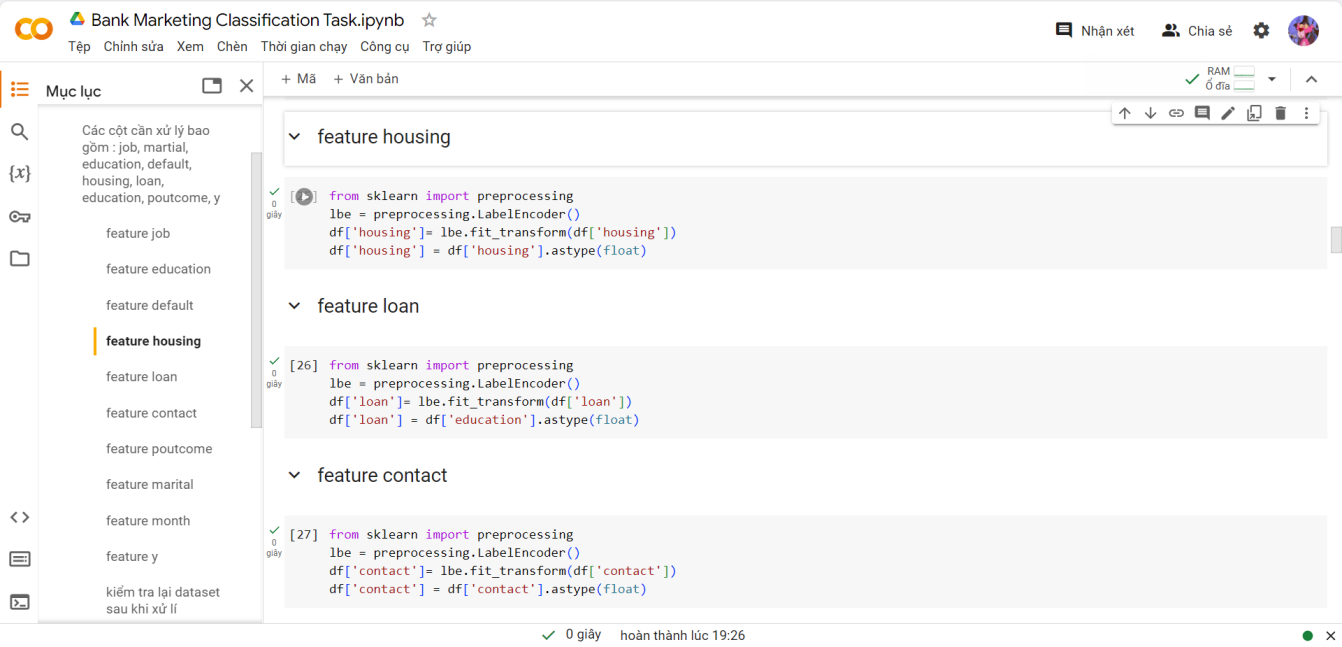
**Bước 3:** Xử lý các cột education, default, housing, loan, education, poutcome, merital, y

Vì các feature này có ít các giá trị phân loại nên ta sẽ xử lý bằng Label Encoder.

Ta sẽ tiến hành gọi Label Encoder và gán vào biến tạm , dùng fit\_transform() để chuyển đổi dữ liệu rồi ghi đè lên dữ liệu cũ, chuyển datatype feature sang định dạng có thể đưa vào bài toán phân loại được.



Hình 8: Xử lý các feature bằng Label Encoder

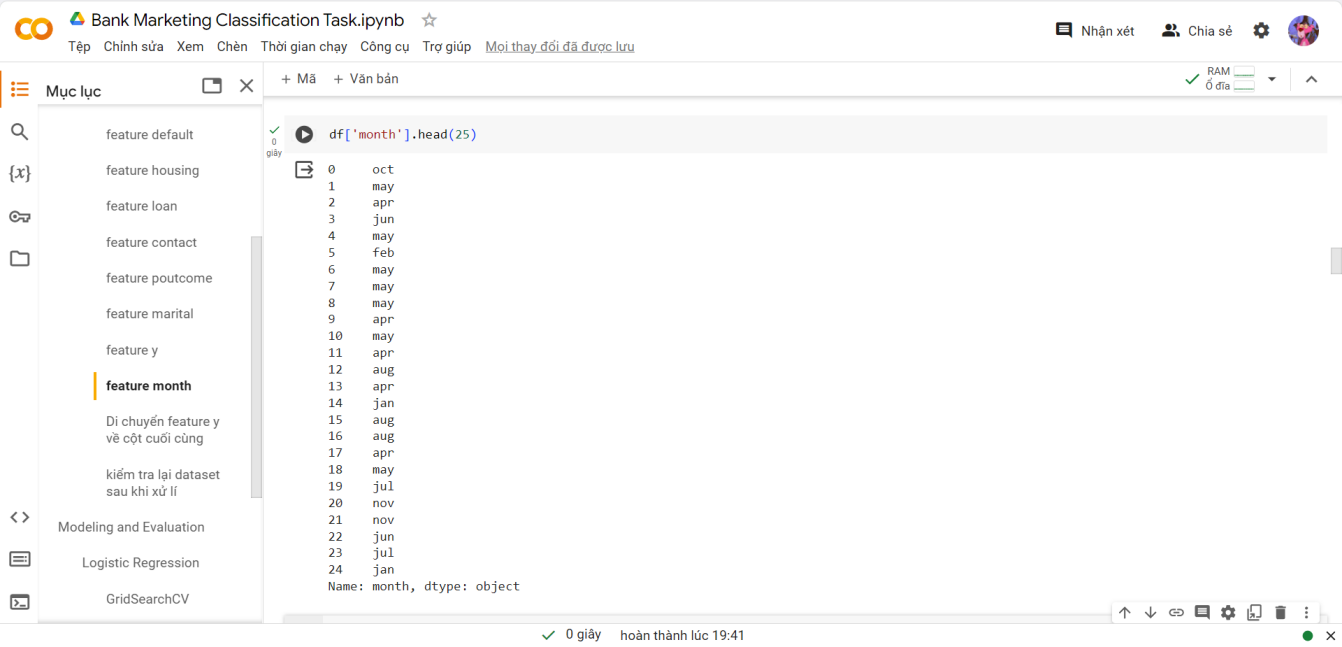


Hình 9: Xử lý các feature bằng Label Encoder



Hình 10: Xử lý các feature bằng Label Encoder

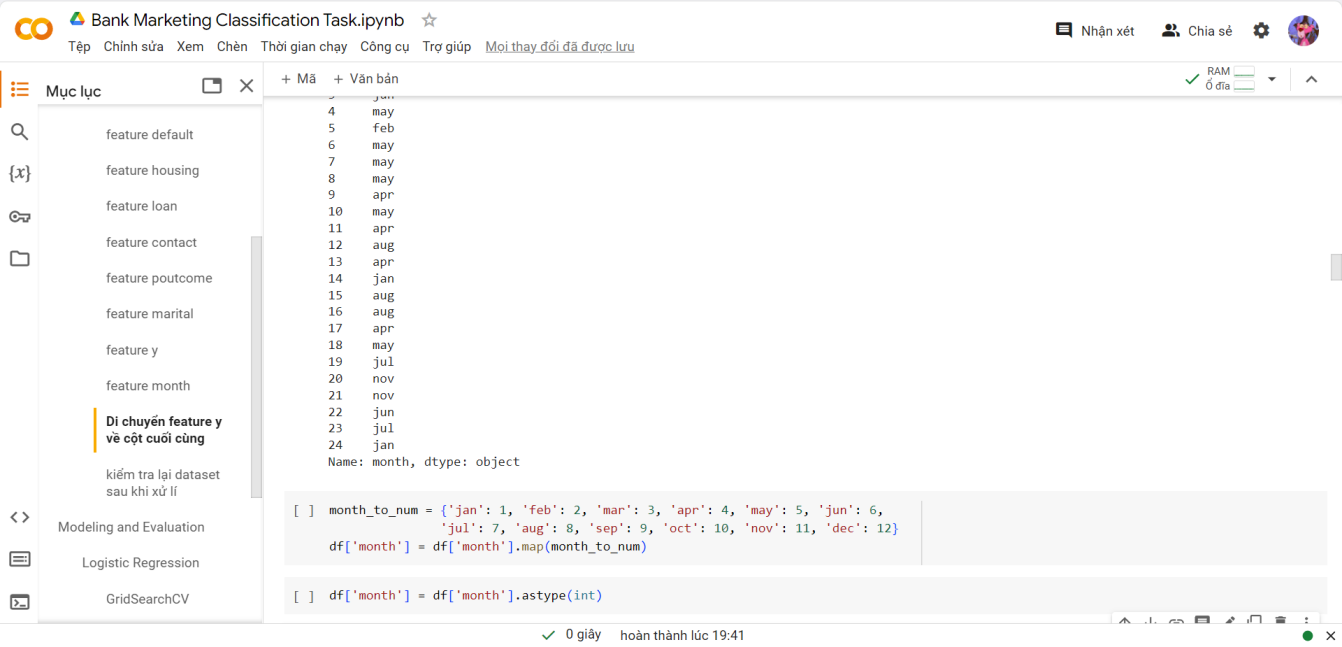
**Bước 4:** Xử lý feature month



Hình 11: Hình ảnh dữ liệu cột month

Trong bộ dữ liệu này, cột month chứa tháng liên lạc cuối cùng được viết tắt bằng 3 ký tự đầu của mỗi tháng (jan là January miêu tả tháng 1).

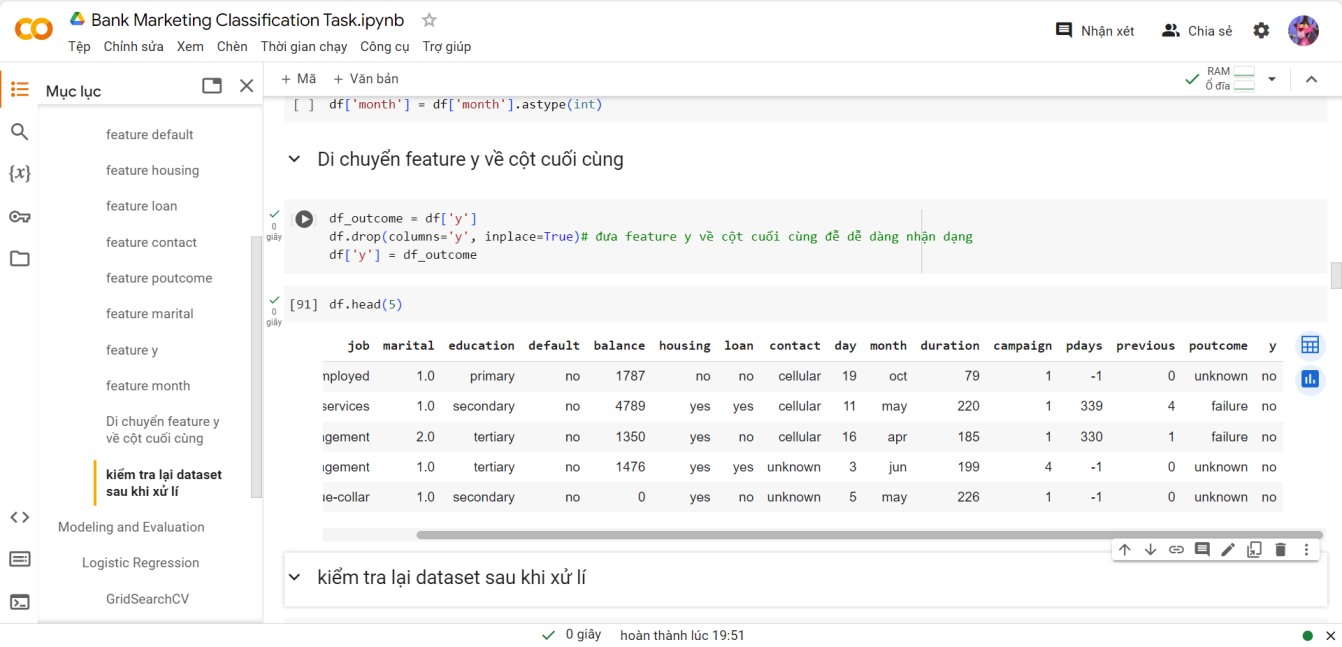
Ở cột này, ta cũng xử lý bằng các gán các giá trị chứa tháng thành số tương ứng thông qua 1 dict được tạo sẵn, sau đó gán giá trị cho từng do dữ liệu thông qua phương thức map(), cuối cùng gán datatype của feature thành dạng float.



Hình 12: Các bước xử lý feature month

**Bước 5:** Tiến hành chuyển y về cột cuối cùng

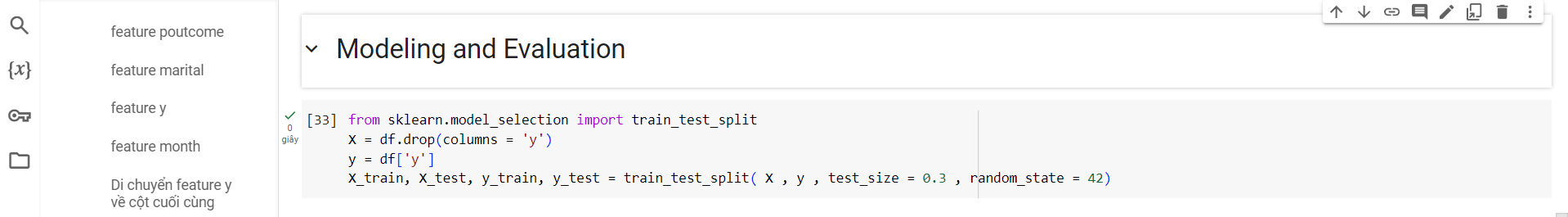
Bước này nhằm dễ dàng xác định feature cần phân loại.



Hình 13: Chuyển feature y về cột cuối cùng

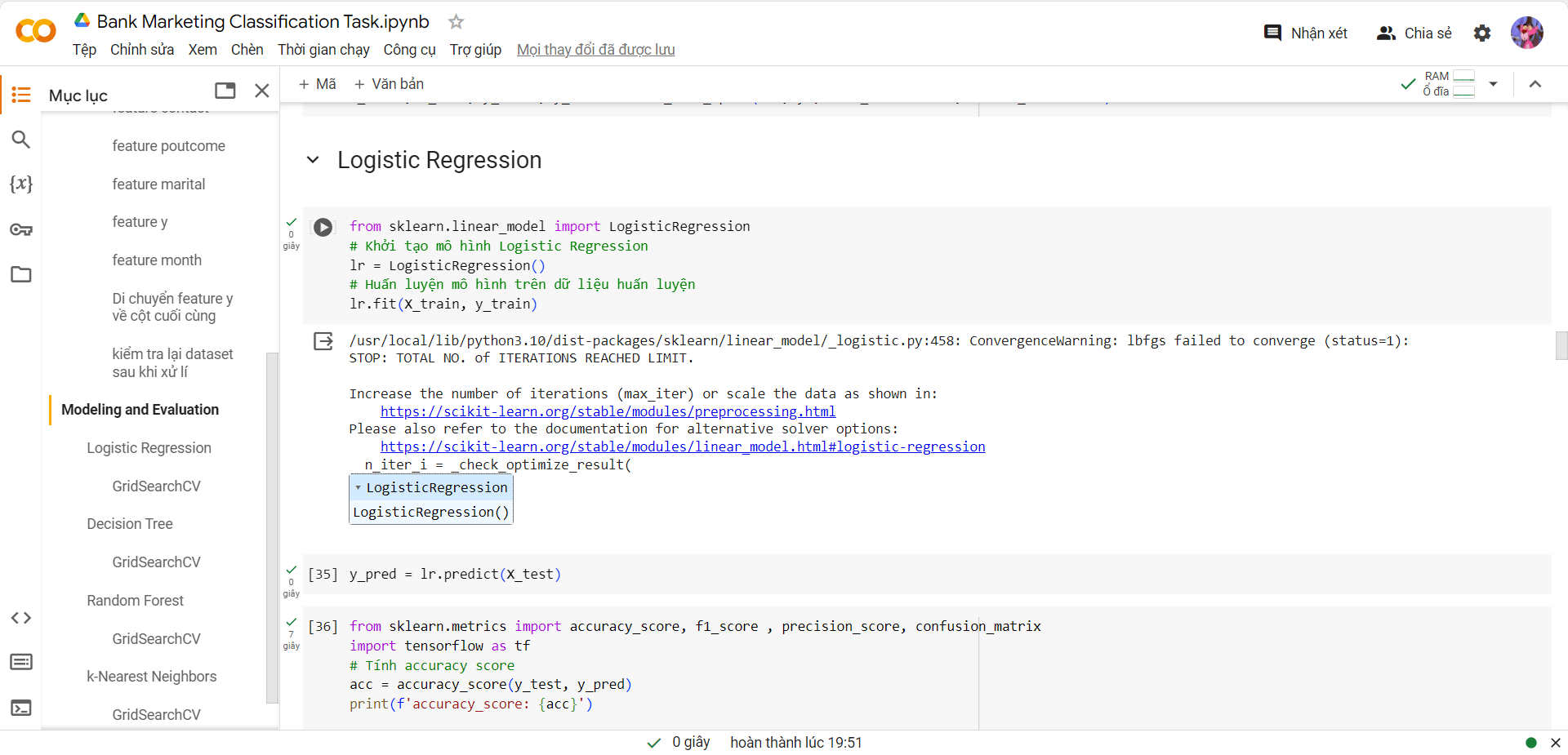
## 4.2. Huấn luyện mô hình và dự đoán

### 4.2.1. Chia dataset thành các tập train và test



Hình 14: Chia tập dữ liệu thành các tập train và test

### 4.2.2. Áp dụng mô hình Logistic Regression



Hình 15: Mô hình Logistic Regression



Hình 16: Sử dụng các metric để đánh giá mô hình



Hình 17: In ra 1 vài kết quả dự đoán đầu tiên



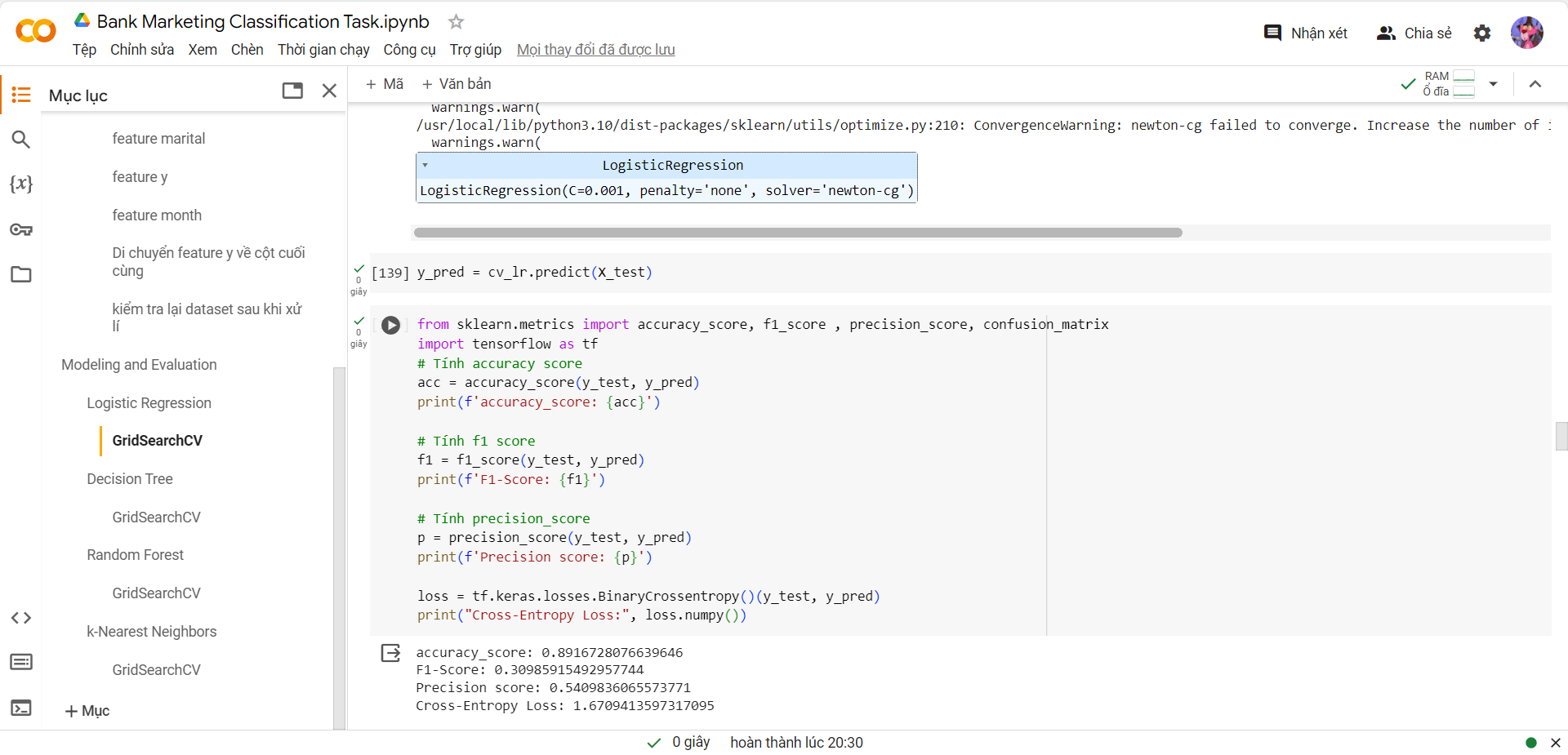
Hình 18: Sử dụng GridSearchCV để tìm các param tốt nhất



Hình 19: Ma trận nhầm lẫn cho mô hình Logistic



Hình 20: Sử dụng các param để huấn luyện lại mô hình



Hình 21: Chạy lại mô hình sau khi dùng GridSearchCV



Hình 23: Ma trận nhầm lẫn mới

### 4.2.3. Áp dụng mô hình Decision Tree



Hình 24: Áp dụng mô hình Decision Tree và đánh giá

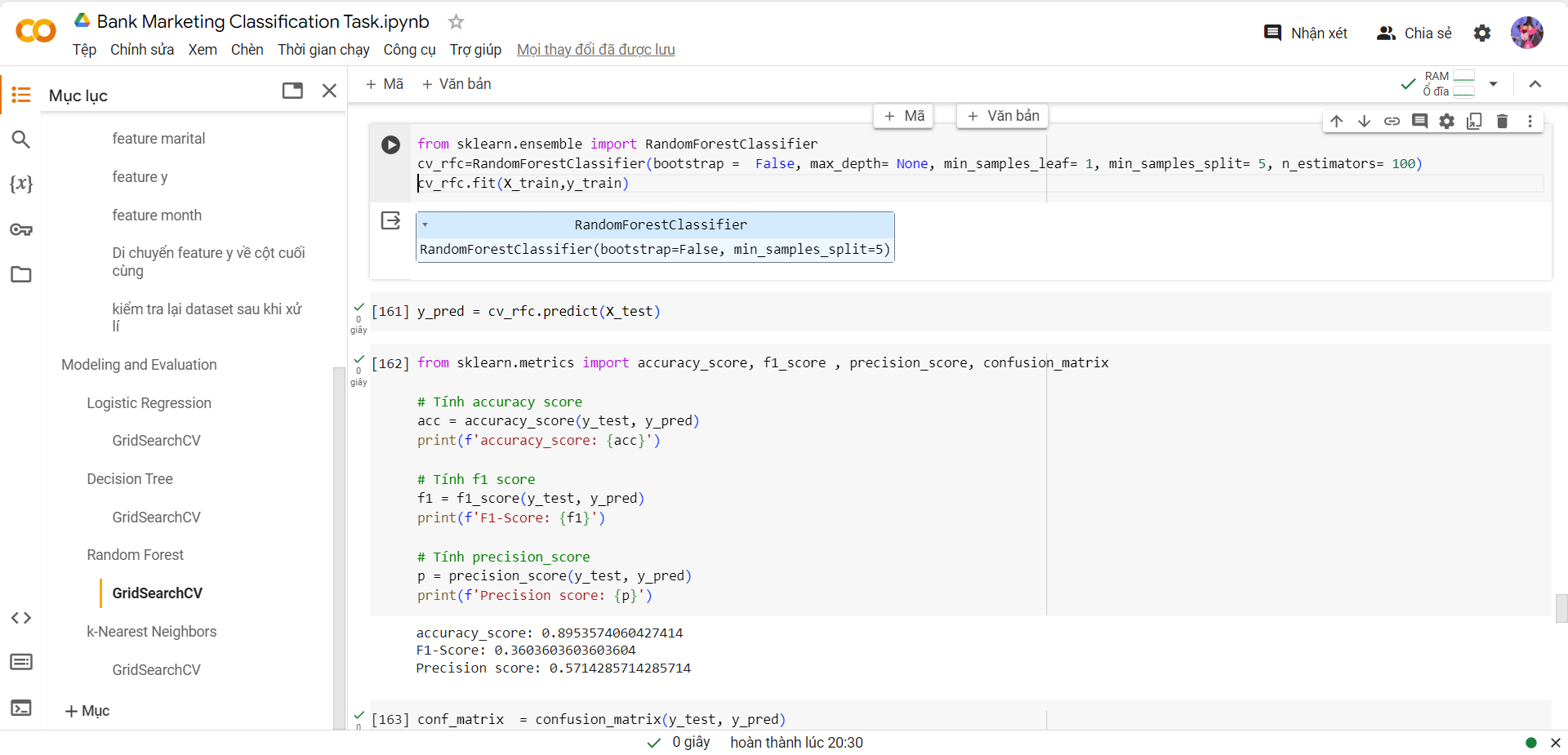


Hình 25: Chạy lại mô hình sau khi dùng GridSearchCV

### 4.2.4 Áp dụng mô hình Random Forest

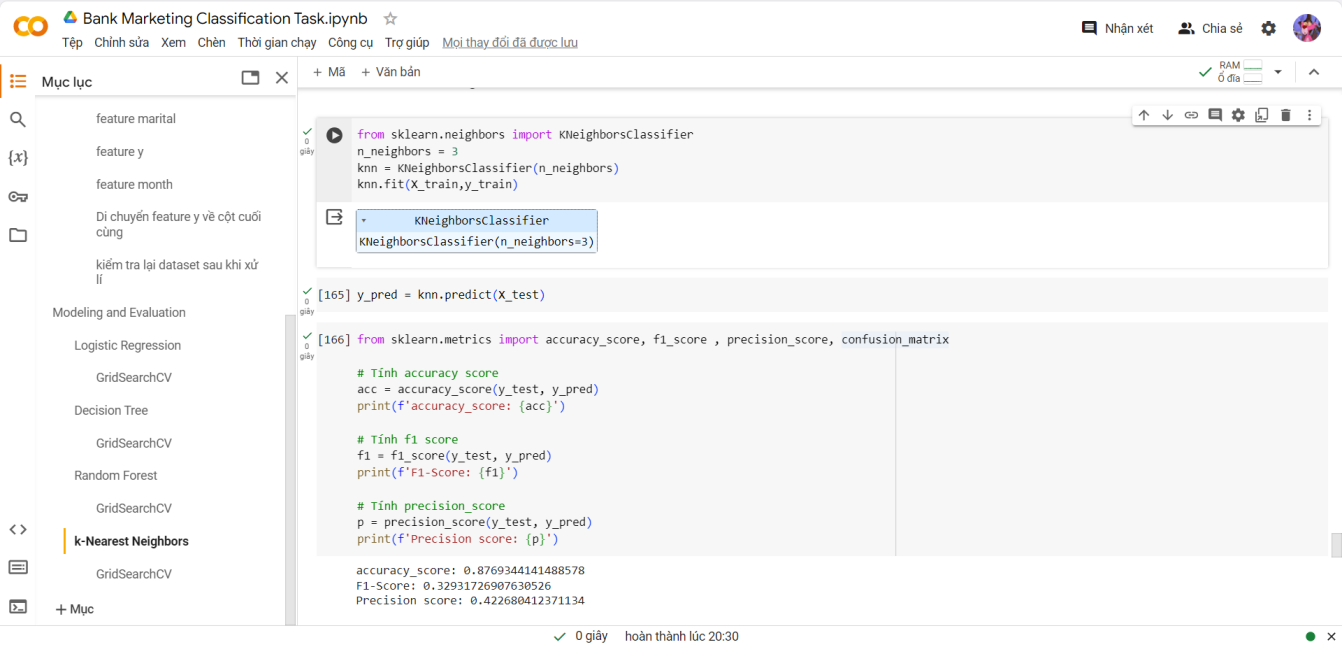


Hình 26: Áp dụng mô hình Random Forest và đánh giá



Hình 27: Chạy lại mô hình sau khi dùng GridSearchCV

### 4.2.5 Áp dụng mô hình k-Nearest Neighbors



Hình 28: Áp dụng mô hình KNN và đánh giá



Hình 29: Chạy lại mô hình sau khi dùng GridSearchCV

## 4.3. Đánh giá mô hình

Linear Regression:

* Accuray Score: Accuracy của mô hình Linear Regression là khá cao, đạt khoảng 0.89, sau khi dùng GridSearchCV và huấn luyện lại mô hình giá trị này vẫn ở mức 0.89. Nhưng đây chưa phải là thước đo chuẩn để đánh mô hình phân loại nên cần có các metric khác để đánh giá thêm.
* F1-Score: F1 có giá trị khoảng 0.306, cho thấy mô hình có hiệu suất dự đoán khá thấp, sau khi dùng GridSearchCV và huấn luyện lại mô hình, giá trị này tăng lên tầm 0.309
* Pecision-Score: Pecision-Score có giá trị khoảng 0.5238, cho thấy mô hình có khả năng dự đoán dương tính thật (TP) nằm ở mức khá, sau khi dùng GridSearchCV và huấn luyện lại mô hình, giá trị này tăng lên tầm 0.54
* Cross Entropy Loss: Cross Entropy Loss có giá trị khoảng 1.69, do cho thấy mô hình dự đoán sai lệch khá nhiều. sau khi dùng GridSearchCV và huấn luyện lại mô hình, giá trị này giảm xuống còn 1.67 cho thấy mô hình đã cải thiện hơn so với lúc trước
* Nhận xét: Logistic Regression có thể không đủ mạnh để mô hình hóa mối quan hệ phức tạp giữa các đặc trưng và đưa ra phân loại. Có thể cần thêm tính linh hoạt với mô hình phức tạp hơn.

Decision Tree:

* Accuray Score: Accuracy của mô hình Decision Tree là khá cao, đạt khoảng 0.867, sau khi dùng GridSearchCV và huấn luyện lại mô hình giá trị này vẫn ở mức 0.867. Nhưng đây chưa phải là thước đo chuẩn để đánh mô hình phân loại nên cần có các metric khác để đánh giá thêm.
* F1-Score: F1 có giá trị khoảng 0.42, cho thấy mô hình có hiệu suất dự đoán khá thấp, sau khi dùng GridSearchCV và huấn luyện lại mô hình, giá trị này vẫn ở mức 0.42.
* Pecision-Score: Pecision-Score có giá trị khoảng 0.4125, cho thấy mô hình có khả năng dự đoán dương tính thật (TP) nằm ở mức khá, sau khi dùng GridSearchCV và huấn luyện lại mô hình, giá trị này tăng lên tầm 0.6.
* Nhận xét: Decision Tree đưa ra phân loại khá ổn. Nhưng vẫn cần áp dụng thêm 1 vài mô hình khác để tìm mô hình tối ưu cho bài toán này.

Random Forest:

* Accuray Score: Accuracy của mô hình Decision Tree là khá cao, đạt khoảng 0.89, sau khi dùng GridSearchCV và huấn luyện lại mô hình giá trị này vẫn ở mức 0.89. Nhưng đây chưa phải là thước đo chuẩn để đánh mô hình phân loại nên cần có các metric khác để đánh giá thêm.
* F1-Score: F1 có giá trị khoảng 0.36, cho thấy mô hình có hiệu suất dự đoán khá thấp, sau khi dùng GridSearchCV và huấn luyện lại mô hình, giá trị này vẫn ở mức 0.36.
* Pecision-Score: Pecision Score có giá trị khoảng 0.6, cho thấy mô hình có khả năng dự đoán dương tính thật (TP) nằm ở mức khá, sau khi dùng GridSearchCV và huấn luyện lại mô hình, giá trị này giảm nhẹ xuống còn 0.57.
* Nhận xét: Random Forest có thể vẫn không đủ mạnh để mô hình hóa mối quan hệ phức tạp giữa các đặc trưng và đưa ra phân loại. Có thể cần thêm tính linh hoạt với mô hình phức tạp hơn.

KNN:

* Accuray Score: Accuracy của mô hình Decision Tree là khá cao, đạt khoảng 0.87, sau khi dùng GridSearchCV và huấn luyện lại mô hình giá trị này vẫn ở mức 0.88. Nhưng đây chưa phải là thước đo chuẩn để đánh mô hình phân loại nên cần có các metric khác để đánh giá thêm.
* F1-Score: F1 có giá trị khoảng 0.33, cho thấy mô hình có hiệu suất dự đoán khá thấp, sau khi dùng GridSearchCV và huấn luyện lại mô hình, giá trị này giảm nhẹ xuống mức 0.28.
* Pecision-Score: Pecision Score có giá trị khoảng 0.42, cho thấy mô hình có khả năng dự đoán dương tính thật (TP) nằm ở mức khá tệ, sau khi dùng GridSearchCV và huấn luyện lại mô hình, giá trị này tăng nhẹ xuống còn 0.5.
* Nhận xét: KNN không phù hợp với bài toán này. Có thể cần phải xem xét các mô hình khác.

Tổng kết lại, dựa trên các kết quả trên, Decision Tree là lựa chọn tốt hơn cho bài toán dự đoán phân loại tiếp thị ngân hàng này. Nhưng kết dự đoán vẫn chưa được tối ưu. Có thể cần phải xem xét lại quá trình tiền xử lý các đặc trưng thì có thể đạt được kết quả dự đoán tốt hơn.

## 4.4. Mã nguồn và tài liệu

Mã nguồn của đoạn code:

<https://drive.google.com/drive/u/0/folders/1Xvd5FhLqimInjs8PvBSYDCsfat99Xx0w>

Mã nguồn của tập dữ liệu:

<https://www.kaggle.com/datasets/adilashrafi/bank-marketing-classification-task>

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết quả đạt được

Qua bài tiểu luận môn học Khai thác dữ liệu và ứng dụng, tôi đã có cơ hội tìm hiểu sâu rộng về các khái niệm và phương pháp trong lĩnh vực này. Cùng với việc áp dụng kiến thức từ Chương 2 về cơ sở lý thuyết vào Chương 4 để ứng dụng thuật toán: Logistic Regression để dự đoán giá tiền cho bất động sản và các phương pháp đánh giá mô hình như Accuracy, F1-Score, Precision-Score, tôi đã thu hoạch được rất nhiều kinh nghiệm quý báu trong việc xử lý và khai tác dữ liệu.

Trong quá trình tiền xử lý dữ liệu ở Chương 4 Mục 4.1 của bài tiểu luận, chúng ta đã thực hiện một loạt các bước để đảm bảo tính chính xác và hiệu quả của dữ liệu một cách có logic và khoa học, quá trình này bao gồm 4 bước:

* Bước 1: Chọn ra những đặc trưng quan trọng
* Bước 2: Xử lý feature job
* Bước 3: Xử lý các cột education, default, housing, loan, education, poutcome, y, marital
* Bước 4: Xử lý feature month

Những bước tiền xử lý dữ liệu trên đã đồng loạt đóng góp vào việc tạo ra một tập dữ liệu sạch sẽ, hiệu quả và chuẩn hóa, sẵn sàng để áp dụng cho việc phân loại ở Chương 4 Mục 4.2 và đưa ra đánh giá.

Kết quả thực nghiệm với Precision-Score khá tốt cho thấy rằng mô hình Decision Tree thực hiện công việc phân loại trên tập dữ liệu này khá tốt. Các giá trị phân loại mang lại hiệu năng ở mức ổn.

Trong quá trình nghiên cứu và thực hiện tiểu luận này, tôi đã tích lũy được nhiều kiến thức và kỹ năng quan trọng, từ khả năng hiểu rõ các thuật toán đến việc áp dụng chúng vào thực tế. Điều này giúp tôi thấy mình đã đóng góp một phần nhỏ nhưng quan trọng vào lĩnh vực này và mở ra những cơ hội mới trong tương lai. Tiểu luận này không chỉ là kết quả của quá trình học tập mà còn là bước đi quan trọng trong sự phát triển cá nhân và sự nghiệp của tôi.

## 5.2. Hạn chế và hướng phát triển của đề tài

Hạn chế của đề tài:

* Mô hình này chỉ áp dụng cho 1 ngân hàng ở Bồ Đào Nha chứ không bao quát hết tất cả.

Về hướng phát triển cho bài toán phân loại tiếp thị ngân hàng, có một số đề xuất như sau:

* Mở Rộng Tập Dữ Liệu: Thu thập thêm dữ liệu từ các nguồn đáng tin cậy như hồ sơ giao dịch, thông tin tài khoản, và dữ liệu liên quan đến hoạt động giao dịch của khách hàng.
* Mở rộng phạm vi thu thập dữ liệu để đảm bảo rằng mô hình có thể hiểu được đa dạng của các chiến lược tiếp thị ngân hàng.
* Tối Ưu Hóa Thêm Tham Số: Nghiên cứu và tối ưu hóa các tham số của mô hình phân loại để tối đa hóa hiệu suất dự đoán. Kiểm tra và đánh giá các phương pháp tối ưu hóa để đạt được sự cân bằng tốt nhất giữa chính xác và khả năng tổng quát hóa trong ngữ cảnh tiếp thị ngân hàng.
* Kết Hợp Các Thuật Toán Khác Nhau: Sử dụng thêm nhiều thuật toán học máy khác nhau như Gradient Boosting, và Neural Networks để tận dụng sức mạnh của mỗi thuật toán. Kết hợp các thuật toán để xây dựng một mô hình phân loại đa dạng và mạnh mẽ.
* Thay Đổi Biến Mục Tiêu: Đưa ra nhiều biến mục tiêu khác nhau như xác định khách hàng có khả năng quan tâm đến một sản phẩm cụ thể hay không, hoặc dự đoán khả năng gặp khó khăn tài chính. Mở rộng ứng dụng của mô hình để dự đoán các chiến lược tiếp thị cụ thể như chiến dịch quảng cáo hiệu quả.
* Phát Triển Giao Diện Người Dùng: Xây dựng một giao diện người dùng linh hoạt và dễ sử dụng để nhân viên ngân hàng có thể tương tác với mô hình một cách thuận tiện.
* Tích hợp khả năng giải thích và tạo báo cáo để giúp những người sử dụng hiểu rõ hơn về quyết định của mô hình.

Tổng kết đề tài, tôi đã đề xuất một phương pháp phân loại tiếp thị ngân hàng với sự chính xác khá tốt. Các hướng phát triển bao gồm mở rộng tập dữ liệu để tăng tính đại diện, tối ưu hóa thêm tham số để cải thiện hiệu suất, kết hợp các thuật toán khác nhau để đa dạng hóa mô hình, và thậm chí thay đổi biến mục tiêu để khám phá nhiều ứng dụng mới. Đồng thời, việc phát triển một giao diện người dùng thân thiện cũng được đề cập để đảm bảo mô hình có thể được triển khai một cách thuận tiện trong thực tế. Đề tài này hứa hẹn sẽ mang lại đóng góp cho lĩnh vực ngân hàng và học máy trong tương lai.

# TÀI LIỆU THAM KHẢO

https://vi.wikipedia.org/

https://aws.amazon.com/vi/what-is/logistic-regression/