BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

ĐỒ ÁN
HỌC PHẦN MÁY HỌC ỨNG DỤNG

Đề tài

XÂY DỰNG MÔ HÌNH DỰ ĐOÁN QUYỀN TRUY CẬP TÀI NGUYÊN CỦA CÔNG TY AMAZON

Nhóm sinh viên thực hiện:

Đỗ Hiếu Nghĩa	B2016985
Trần Công Nhật	B2016989
Diệp Nguyễn Minh Tuyến	B2017016

Giảng viên hướng dẫn:
TS. Mã Trường Thành

Cần Thơ, 11/2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

ĐỒ ÁN
HỌC PHẦN MÁY HỌC ỨNG DỤNG

Đề tài

XÂY DỰNG MÔ HÌNH DỰ ĐOÁN QUYỀN TRUY CẬP TÀI NGUYÊN CỦA CÔNG TY AMAZON

Nhóm sinh viên thực hiện:

Đỗ Hiếu Nghĩa	B2016985
Trần Công Nhật	B2016989
Diệp Nguyễn Minh Tuyến	B2017016

Giảng viên hướng dẫn:
TS. Mã Trường Thành

Cần Thơ, 11/2023

NHẬN XÉT CỦA GIẢNG VIÊN

Cần Thơ, ngày tháng năm
(Ký và ghi rõ họ tên)

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC BẢNG

PHÂN CÔNG CÔNG VIỆC

Bảng 1: Bảng phân công công việc

STT	Họ và tên	Công việc
1	Đỗ Hiếu Nghĩa (Nhóm trưởng)	1. Thu thập dữ liệu 2. Đọc hiểu dữ liệu 3. Tiền xử lý dữ liệu 4. Huấn luyện mô hình (Nghi thức Hold out, KNN, Naïve Bayes, Decision Tree, Random Forest) 5. Triển khai mô hình thành ứng dụng 6. Soạn slide báo cáo
2	Trần Công Nhật	1. Thu thập dữ liệu 2. Đọc hiểu dữ liệu 3. Huấn luyện mô hình (Nghi thức K-fold Decision Tree, Random Forest) 4. Triển khai mô hình thành ứng dụng
3	Diệp Nguyễn Minh Tuyến (Thư ký)	1. Thu thập dữ liệu 2. Đọc hiểu dữ liệu 3. Huấn luyện mô hình (Nghi thức K-fold KNN, Naïve Bayes) 4. Tổng hợp kết quả đánh giá mô hình 5. Viết bài báo cáo

PHẦN NỘI DUNG
CHƯƠNG I. TRỰC QUAN HÓA, XỬ LÝ TẬP DỮ LIỆU

1. Tổng quan về dữ liệu

1.1. Mô tả dữ liệu

- Dữ liệu có tên: Amazon_employee_access
- Nguồn: OpenML ID 4135

Mô tả: Dữ liệu bao gồm 32769 hàng và 11 cột, bộ dữ liệu bao gồm dữ liệu lịch sử thực được thu thập từ năm 2010 và 2011. Nhân viên được cho phép hoặc từ chối quyền truy cập vào tài nguyên theo thời gian. Dữ liệu được sử dụng để tạo ra một thuật toán có khả năng học hỏi từ dữ liệu lịch sử này để dự đoán sự chấp thuận/từ chối cho một nhóm nhân viên chưa từng thấy.

id	RESOURCE	MGR_ID	ROLE_ROLLUP_1	ROLE_ROLLUP_2	ROLE_DEPTNAME	ROLE_TITLE	ROLE_FAMILY_DESC	ROLE_FAMILY	ROLE_CODE	target
1	39353	85475	117961	118300	123472	117905	117906	290919	117908	1
2	17183	1540	117961	118343	123125	118536	118536	308574	118539	1
3	36724	14457	118219	118220	117884	117879	267952	19721	117880	1
4	36135	5396	117961	118343	119993	118321	240983	290919	118322	1
5	42680	5905	117929	117930	119569	119323	123932	19793	119325	1
6	45333	14561	117951	117952	118008	118568	118568	19721	118570	0
7	25993	17227	117961	118343	123476	118980	301534	118295	118982	1
8	19666	4209	117961	117969	118910	126820	269034	118638	126822	1
9	31246	783	117961	118413	120584	128230	302830	4673	128231	1
10	78766	56683	118079	118080	117878	117879	304519	19721	117880	1
11	4675	3005	117961	118413	118481	118784	117906	290919	118786	1
12	15030	94005	117902	118041	119238	119093	138522	119095	119096	1
13	79954	46608	118315	118463	122636	120773	123148	118960	120774	1
14	4675	50997	91261	118026	118202	119962	168365	118205	119964	1
15	95836	18181	117961	118343	118514	118321	117906	290919	118322	1

Hình 1: Dataset Amazon employee access

1.2. Ý nghĩa của dữ liệu

Bộ dữ liệu chứa các thuộc tính sau:

- id: số thứ tự trường dữ liệu.
- RESOURCE: đây là ID cho từng tài nguyên.
- MGR_ID: đây là ID NHÂN VIÊN của người quản lý của bản ghi ID NHÂN VIÊN hiện tại. Một nhân viên chỉ có thể có một người quản lý tại một thời điểm.
- ROLE_ROLLUP_1: Đây là ID danh mục nhóm vai trò của công ty 1.
- ROLE_ROLLUP_2: Đây là ID danh mục nhóm vai trò của công ty 2.
- ROLE_DEPTNAME: Đây là mô tả phòng ban theo vai trò của công ty.
- ROLE_TITLE: Đây là mô tả tiêu đề doanh nghiệp theo vai trò của công ty.
- ROLE_FAMILY_DESC: Đây là mô tả mở rộng về họ vai trò của công ty.
- ROLE_FAMILY: Đây là mô tả về họ vai trò của công ty.

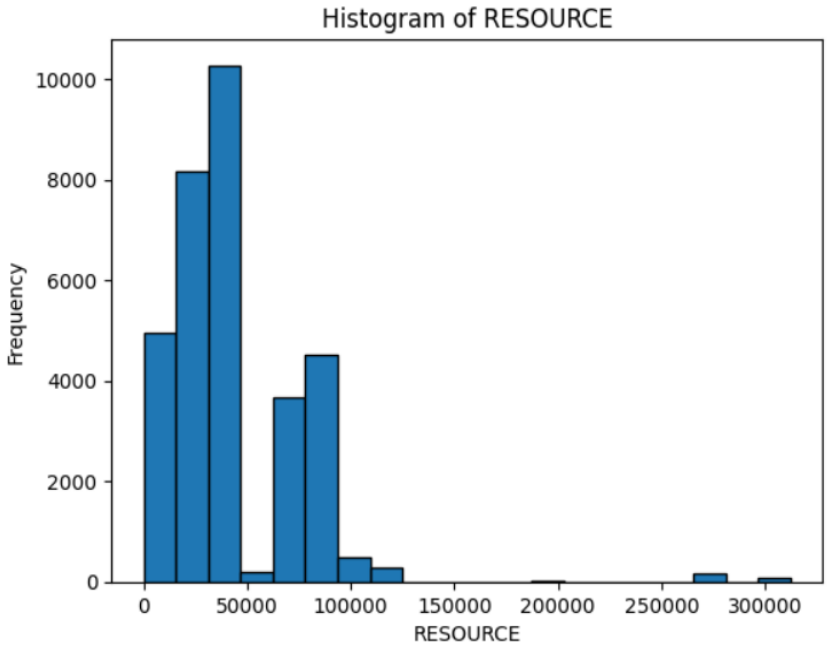
- ROLE_CODE: Đây là mã vai trò của công ty. Mã này là duy nhất cho từng vai trò.
- target (nhãn): Giá trị 1 hoặc 0 tương ứng với quyền truy cập được từ chối hay không?

Bộ dữ liệu là một nguồn tài nguyên có giá trị để đào tạo các mô hình học máy để dự đoán sự chấp thuận/từ chối quyền truy cập của nhân viên. Bộ dữ liệu được ghi chép rõ ràng và chứa nhiều tính năng có thể được sử dụng để đưa ra dự đoán chính xác.

2. Phân tích dữ liệu và tiền xử lý dữ liệu

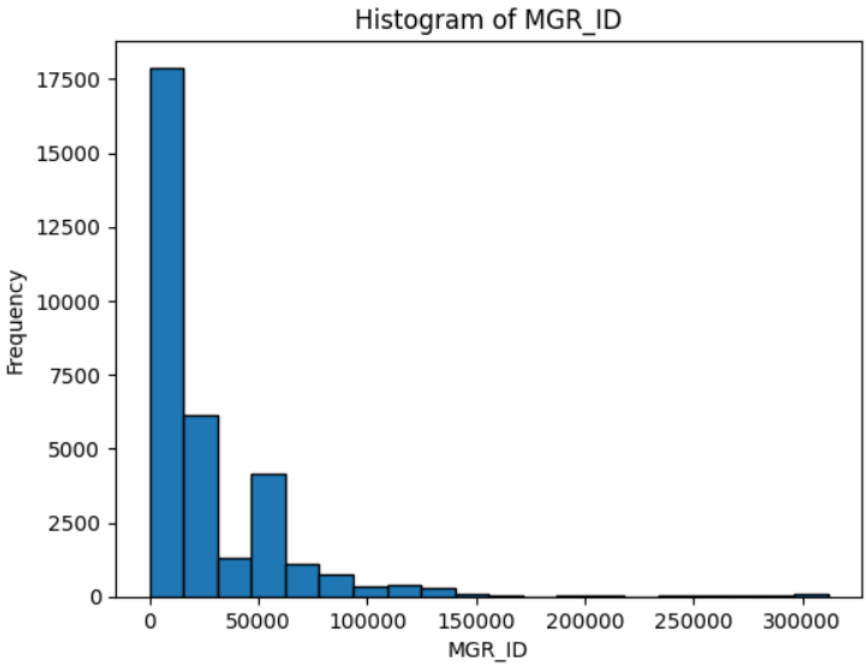
2.1. Phân tích dữ liệu

- RESOURCE



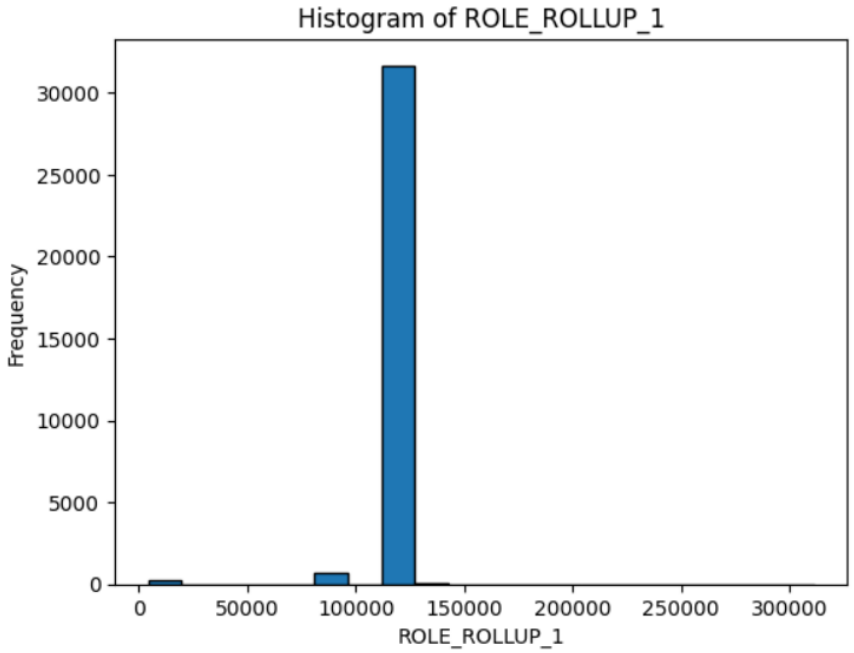
Hình 2: Histogram của thuộc tính RESOURCE

- MGR_ID



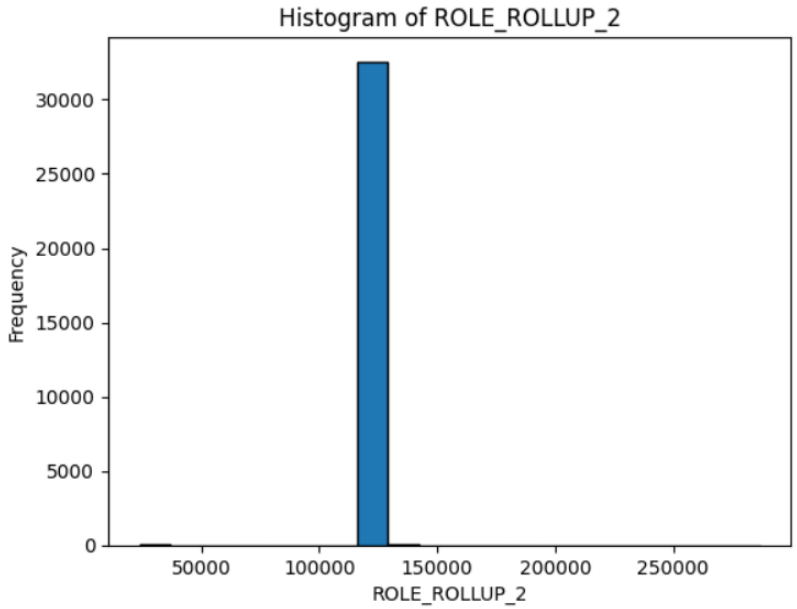
Hình 3: Histogram của thuộc tính MGR_ID

- ROLE_ROLLUP_1



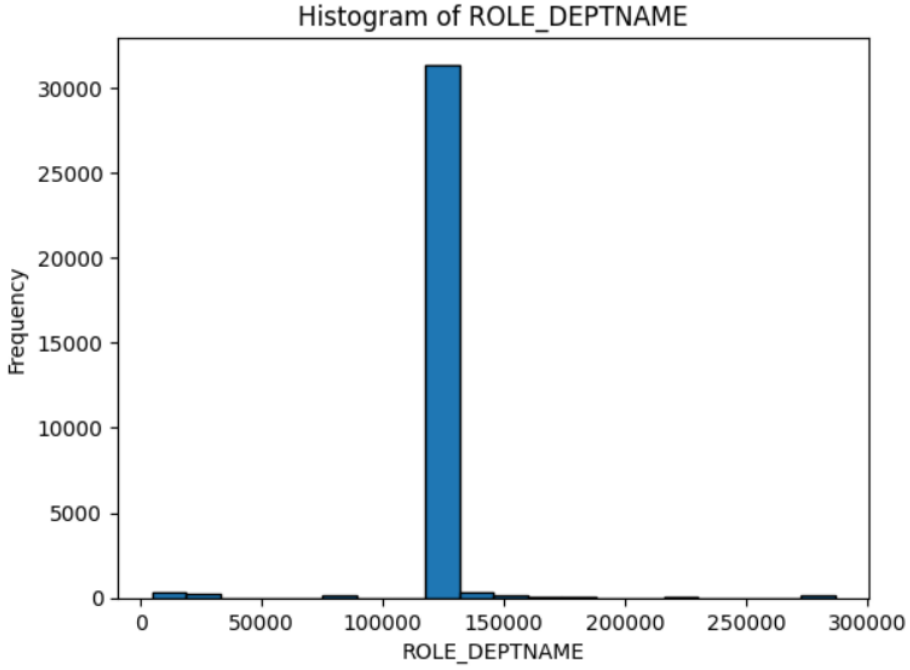
Hình 4: Histogram của thuộc tính ROLE_ROLLUP_1

- ROLE_ROLLUP_2



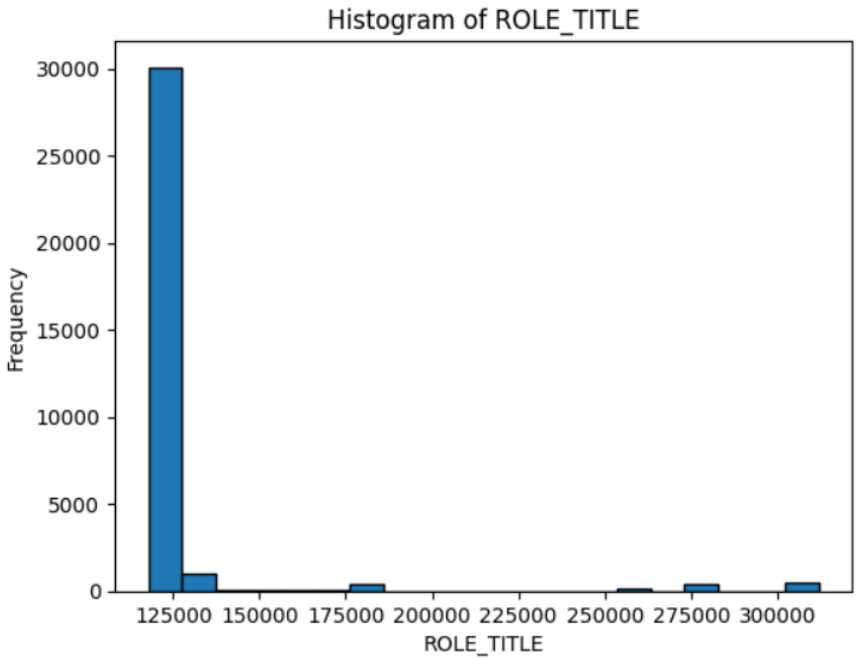
Hình 5: Histogram của thuộc tính ROLE_ROLLUP_2

- ROLE_DEPTNAME



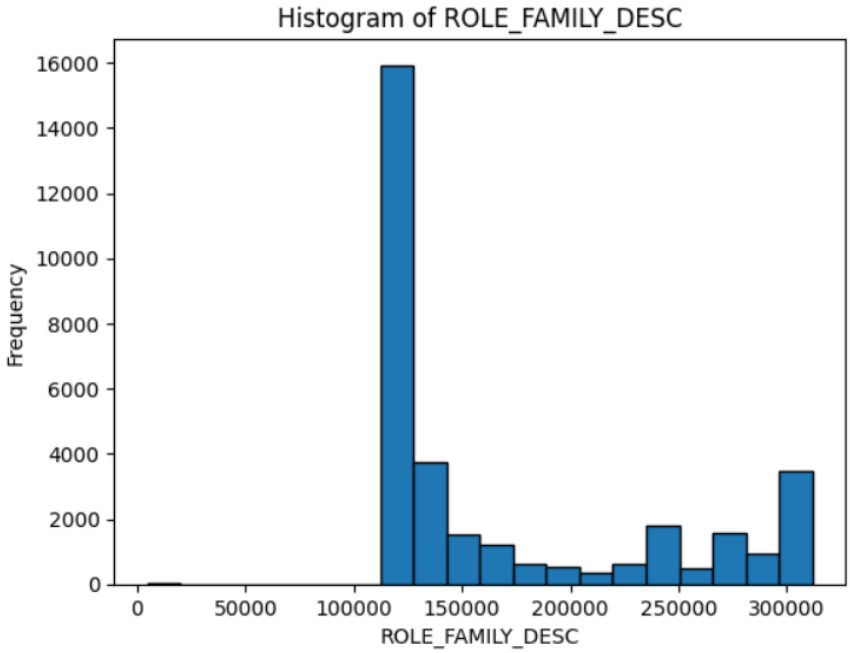
Hình 6: Histogram của thuộc tính ROLE_DEPTNAME

- ROLE_TITLE



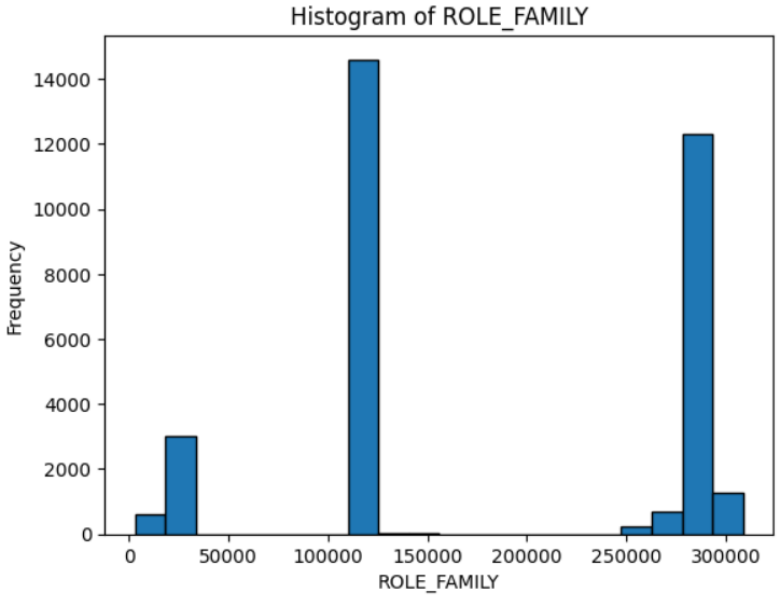
Hình 7: Histogram của thuộc tính ROLE_TITLE

- ROLE_FAMILY_DESC



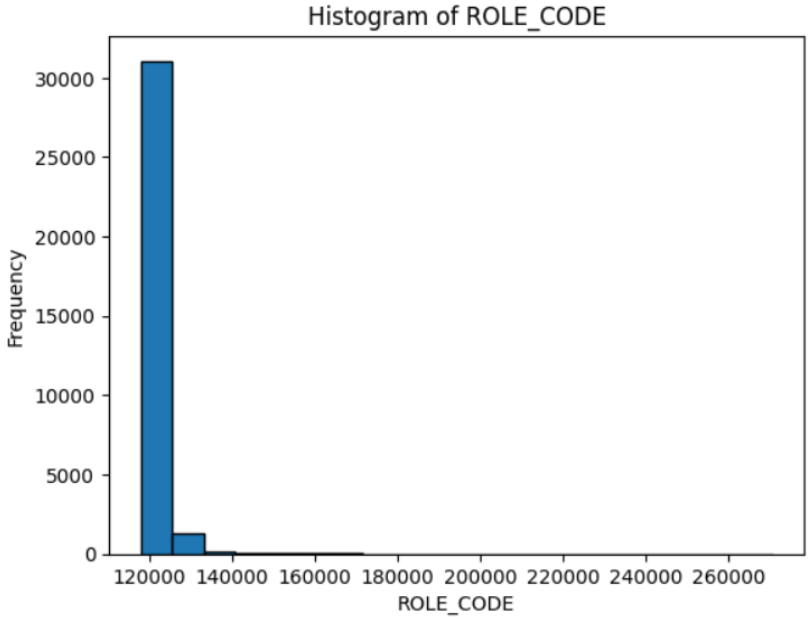
Hình 8: Histogram của thuộc tính ROLE_FAMILY_DESC

- ROLE_FAMILY



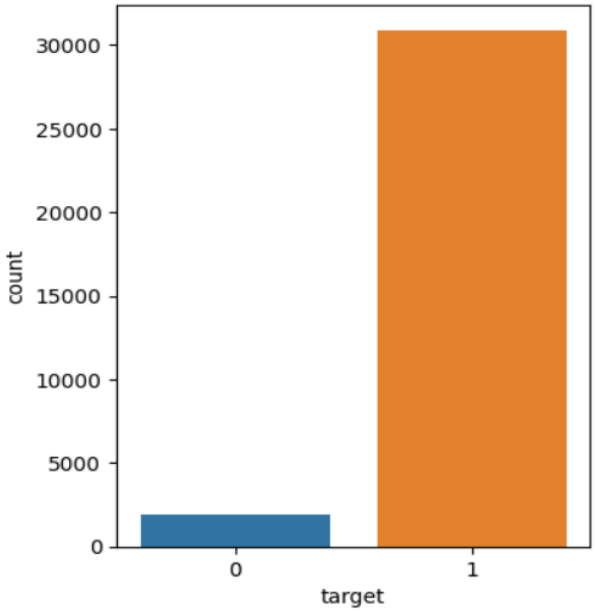
Hình 9: Histogram của thuộc tính ROLE_FAMILY

- ROLE_CODE



Hình 10: Histogram của thuộc tính ROLE_CODE

- Cột nhãn (target)



Nhận xét: có sự mất cấn bằng dữ liệu ở cột nhãn

2.2. Tiền xử lý dữ liệu

- **Kiểm tra ô trống:** dữ liệu không chứa ô trống.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32769 entries, 0 to 32768
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   id                32769 non-null  int64
 1   RESOURCE          32769 non-null  int64
 2   MGR_ID            32769 non-null  int64
 3   ROLE_ROLLUP_1     32769 non-null  int64
 4   ROLE_ROLLUP_2     32769 non-null  int64
 5   ROLE_DEPTNAME     32769 non-null  int64
 6   ROLE_TITLE        32769 non-null  int64
 7   ROLE_FAMILY_DESC  32769 non-null  int64
 8   ROLE_FAMILY       32769 non-null  int64
 9   ROLE_CODE         32769 non-null  int64
 10  target            32769 non-null  int64
dtypes: int64(11)
memory usage: 2.8 MB
```

Hình 11: kết quả tóm tắt thông tin về dataframe

- **Xóa thuộc tính không cần thiết:** xóa cột id, cột id là số thứ tự không có mối tương quan với các thuộc tính khác.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32769 entries, 0 to 32768
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   RESOURCE         32769 non-null  int64
 1   MGR_ID           32769 non-null  int64
 2   ROLE_ROLLUP_1    32769 non-null  int64
 3   ROLE_ROLLUP_2    32769 non-null  int64
 4   ROLE_DEPTNAME    32769 non-null  int64
 5   ROLE_TITLE       32769 non-null  int64
 6   ROLE_FAMILY_DESC 32769 non-null  int64
 7   ROLE_FAMILY      32769 non-null  int64
 8   ROLE_CODE        32769 non-null  int64
 9   target           32769 non-null  int64
dtypes: int64(10)
memory usage: 2.5 MB
```

Hình 12: kết quả dataframe sau khi xóa cột id

- **Kiểm tra các mẫu bị trùng lặp:** không có mẫu nào bị trùng lặp.

```
Số lượng giá trị trùng lặp: 0
0        False
1        False
2        False
3        False
4        False
         ...
32764    False
32765    False
32766    False
32767    False
32768    False
Length: 32769, dtype: bool
```

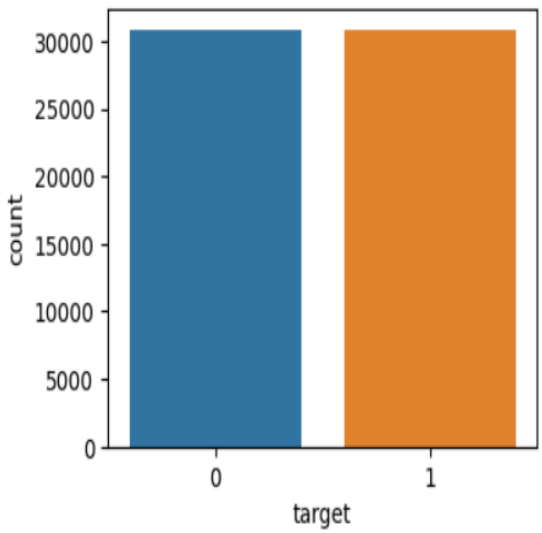
Hình 13: kết quả kiểm tra các mẫu có bị trùng lặp không?

- **Cân bằng dữ liệu:**

Mất cân bằng dữ liệu có thể dẫn đến các vấn đề như hiệu suất kém trên lớp thiểu số hoặc phương sai mô hình cao => Sử dụng phương pháp SMOTE để cân bằng lại dữ liệu.

SMOTE (Synthetic Minority Over-sampling Technique) là một phương pháp oversampling (tăng cường) dữ liệu được sử dụng để cân bằng mẫu giữa các lớp trong bài toán phân loại, đặc biệt là khi dữ liệu là mất cân bằng (imbalanced). SMOTE tạo ra các mẫu nhân tạo cho lớp thiểu số bằng cách kết hợp thông tin từ các mẫu hiện có của lớp đó. Điều này có thể giúp mô hình học tốt hơn trên lớp thiểu số và cải thiện hiệu suất của mô hình trong tình huống mất cân bằng.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 61744 entries, 0 to 61743
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   RESOURCE          61744 non-null  int64
 1   MGR_ID            61744 non-null  int64
 2   ROLE_ROLLUP_1     61744 non-null  int64
 3   ROLE_ROLLUP_2     61744 non-null  int64
 4   ROLE_DEPTNAME     61744 non-null  int64
 5   ROLE_TITLE        61744 non-null  int64
 6   ROLE_FAMILY_DESC  61744 non-null  int64
 7   ROLE_FAMILY       61744 non-null  int64
 8   ROLE_CODE         61744 non-null  int64
 9   target            61744 non-null  int64
dtypes: int64(10)
memory usage: 4.7 MB
```



Hình 14: kết quả dataframe sau khi cân bằng lại dữ liệu

- **Chuẩn hóa dữ liệu:** sử dụng phương pháp Min Max Scaling.

RESOURCE	MGR_ID	ROLE_ROLLUP_1	ROLE_ROLLUP_2	ROLE_DEPTNAME	ROLE_TITLE	ROLE_FAMILY_DESC	ROLE_FAMILY	ROLE_CODE	target
0.126069588	0.274167311	0.37039487	0.35937904	0.421093301	0.000134029	0.368604205	0.942198897	0.000183233	1
0.055046724	0.004860895	0.37039487	0.35954253	0.41986332	0.003386807	0.370655026	1	0.004312517	1
0.117647436	0.046305239	0.371235573	0.359074871	0.401285987	0	0.857044734	0.054317649	0	1
0.115760541	0.017232915	0.37039487	0.35954253	0.408761582	0.002278491	0.769253306	0.942198897	0.002892462	1
0.136727823	0.018866048	0.370290597	0.35797226	0.407258665	0.007443759	0.388220473	0.054553371	0.009456126	1
0.14522686	0.046638924	0.370362284	0.358055906	0.401725519	0.003551766	0.370759195	0.054317649	0.004515382	0
0.083270063	0.055192816	0.37039487	0.35954253	0.42110748	0.005675609	0.966363275	0.377041291	0.007211523	1
0.063001156	0.013424412	0.37039487	0.358120542	0.404922763	0.04609048	0.860566938	0.378164246	0.05851673	1
0.100098349	0.002432052	0.37039487	0.359808678	0.41085645	0.053358971	0.970582108	0.005051662	0.06773727	1
0.252331389	0.181787847	0.370779377	0.358542576	0.401264719	0	0.976080262	0.054317649	0	1
0.01497663	0.009561364	0.37039487	0.359808678	0.403402123	0.004665237	0.368604205	0.942198897	0.005928893	1
0.048149465	0.301535914	0.370202616	0.358394294	0.406085397	0.006258119	0.43571489	0.379660429	0.007957542	1
0.256137215	0.149462093	0.371548393	0.359998783	0.418130002	0.014918449	0.38566834	0.379218449	0.018938427	1
0.01497663	0.16354425	0.283391878	0.358337262	0.402413175	0.010737778	0.53286197	0.376746638	0.013637762	1

Hình 15: dataset sau khi chuẩn hóa dữ liệu

- Hình 15 là kết quả dataset sau giai đoạn tiền xử lý dữ liệu.

3. Cấu hình máy tính huấn luyện mô hình

❖ Máy tính tiến hành huấn luyện và đánh giá mô hình

- Công nghệ sử dụng: Google Colab
- GPU thường là GPU Tesla K80
- Bộ nhớ GDDR5: 12 GB
- Tốc độ xung nhịp: 1.4 GHz

❖ Máy tính tiến hành triển khai mô hình thành ứng dụng

Bảng 2: Bảng cấu hình máy tính triển khai mô hình

	Phần cứng
Máy 1	CPU: Intel Core i7 1165G RAM: 8GB Ổ cứng: 512GB SSD
Máy 2	CPU: Intel Core(TM) i7-1065G7 RAM: 16GB Ổ cứng: 512GB SSD

CHƯƠNG II. HUẤN LUYỆN MÔ HÌNH

1. Giải thuật KNN

1.1. Giới thiệu

Giải thuật KNN (K-Nearest Neighbors) là một thuật toán học máy có giám sát, được sử dụng để phân loại dữ liệu mới dựa trên thông tin của các dữ liệu đã được phân loại trước đó. Thuật toán này dựa trên giả thuyết rằng các dữ liệu tương tự nhau sẽ nằm gần nhau trong không gian đặc trưng.

1.2. Cách hoạt động của giải thuật

Giải thuật KNN hoạt động như sau:

- Đầu tiên, thuật toán sẽ xây dựng một tập huấn luyện gồm các dữ liệu đã được phân loại.
- Khi có một dữ liệu mới cần được phân loại, thuật toán sẽ tính khoảng cách giữa dữ liệu mới đó và tất cả các dữ liệu trong tập huấn luyện.
- Sau đó, thuật toán sẽ tìm ra K dữ liệu gần nhất với dữ liệu mới.
- Cuối cùng, thuật toán sẽ gán nhãn cho dữ liệu mới bằng nhãn của K dữ liệu gần nhất.

1.3. Ưu điểm, nhược điểm của giải thuật

❖ **Ưu điểm:**

- Đơn giản, dễ hiểu, dễ triển khai.
- Không cần thiết phải xây dựng mô hình phức tạp.
- Có thể áp dụng cho các tập dữ liệu có kích thước lớn.

❖ **Nhược điểm:**

- Độ chính xác của thuật toán phụ thuộc vào tham số k.
- Có thể bị ảnh hưởng bởi các giá trị ngoại lệ.

1.4. Kết quả huấn luyện mô hình

- **Kết quả huấn luyện mô hình KNN theo nghi thức Hold-out, datatrain 80% và datatest 20%**

Bảng 3: Bảng kết quả huấn luyện mô hình KNN nghi thức hold-out

		Precision	Recall	F1 score	Training time	Inference time
KNN	K = 3	0.906	0.9	0.899	0.095	1.27
	K = 5	0.893	0.882	0.882	0.092	1.277
	K = 7	0.883	0.871	0.871	0.086	1.41
	K = 9	0.876	0.862	0.861	0.095	1.44
	K = 11	0.864	0.849	0.848	0.091	1.41

- **Kết quả huấn luyện mô hình KNN theo nghi thức K-fold, K = 15**

Bảng 4: Bảng kết quả huấn luyện mô hình KNN nghi thức 15-fold

		Precision	Recall	F1 score	Training time	Inference time
KNN	K = 3	0.917	0.912	0.912	0.126	0.471
	K = 5	0.903	0.895	0.894	0.121	0.534
	K = 7	0.892	0.882	0.881	0.114	0.468
	K = 9	0.883	0.871	0.87	0.081	0.394
	K = 11	0.876	0.862	0.861	0.15	0.596

2. Giải thuật Naïve Bayes

2.1. Giới thiệu

Giải thuật Naive Bayes là một thuật toán phân loại dựa trên xác suất, sử dụng định lý Bayes để tính xác suất của một điểm dữ liệu thuộc một lớp nhất định. Thuật toán này dựa trên giả định rằng các thuộc tính của một điểm dữ liệu là độc lập với nhau.

2.2. Cách hoạt động của giải thuật

Giải thuật Naïve Bayes hoạt động như sau:

- Đầu tiên, thuật toán sẽ xây dựng một tập huấn luyện gồm các điểm dữ liệu đã được phân loại.
- Khi có một điểm dữ liệu mới cần được phân loại, thuật toán sẽ tính xác suất của điểm dữ liệu đó thuộc từng lớp.

- Cuối cùng, thuật toán sẽ gán nhãn cho điểm dữ liệu mới bằng lớp có xác suất cao nhất.

2.3. Ưu điểm, nhược điểm của giải thuật

- **Ưu điểm:**
  - Đơn giản, dễ hiểu, dễ triển khai.
  - Có thể áp dụng cho các tập dữ liệu có kích thước lớn.
  - Có thể kết hợp với các kỹ thuật khác để cải thiện độ chính xác.
- **Nhược điểm:**
  - Giả định độc lập giữa các thuộc tính có thể không chính xác trong một số trường hợp.
  - Độ chính xác của thuật toán có thể bị ảnh hưởng bởi các thuộc tính có giá trị hiếm.

2.4. Kết quả huấn luyện mô hình

- **Kết quả huấn luyện mô hình Naïve Bayes theo nghi thức Hold-out, datatrain 80% và datatest 20%**

Bảng 5: Bảng kết quả huấn luyện mô hình Naïve Bayes nghi thức hold-out

	Precision	Recall	F1 score	Training time	Inference time
Naïve Bayes	0.547	0.518	0.434	0.026	0.005

- **Kết quả huấn luyện mô hình Naïve Bayes theo nghi thức K-fold, K = 15**

Bảng 6: Bảng kết quả huấn luyện mô hình Naïve Bayes nghi thức 15-fold

	Precision	Recall	F1 score	Training time	Inference time
Naïve Bayes	0.551	0.519	0.432	0.012	0.001

3. Giải thuật Decision Tree

3.1. Giới thiệu

Giải thuật Decision Tree là một thuật toán học máy có giám sát, được sử dụng để phân loại dữ liệu. Thuật toán này xây dựng một cây quyết định, trong đó mỗi nút đại diện cho một thuộc tính của dữ liệu và mỗi nhánh đại diện cho một giá trị có thể có của thuộc tính đó.

3.2. Cách hoạt động của giải thuật

Giải thuật Decision Tree hoạt động như sau:

- Đầu tiên, thuật toán sẽ xây dựng một tập huấn luyện gồm các điểm dữ liệu đã được phân loại.
- Sau đó, thuật toán sẽ bắt đầu từ nút gốc của cây quyết định và sẽ phân chia dữ liệu trong nút đó thành hai hoặc nhiều nhánh.
- Tiếp theo, thuật toán sẽ lặp lại quy trình này cho mỗi nhánh, cho đến khi tất cả dữ liệu trong các nhánh đều được phân loại.

3.3. Ưu điểm, nhược điểm của giải thuật

❖ **Ưu điểm:**
- Dễ hiểu và dễ giải thích.
- Có thể áp dụng cho nhiều loại dữ liệu khác nhau.
- Có thể được áp dụng cho cả bài toán phân lớp và hồi quy.

❖ **Nhược điểm:**
- Có thể bị quá phức tạp, dẫn đến overfitting.
- Có thể bị ảnh hưởng bởi các dữ liệu ngoại lệ.

3.4. Kết quả huấn luyện mô hình

❖ **Kết quả huấn luyện mô hình Decision Tree theo nghi thức Hold-out, datatrain 80% và datatest 20%**

Bảng 7: Bảng kết quả huấn luyện mô hình Decision Tree nghi thức Hold-out

		Precision	Recall	F1 score	Training time	Inference time
Decision Tree	criterion='gini', max_depth=100, min_samples_split=3, min_samples_leaf=6, random_state=42	0.931	0.931	0.931	0.364	0.004
	criterion='gini', max_depth=200, min_samples_split=3, min_samples_leaf=6, random_state=42	0.93	0.93	0.93	0.851	0.011
	criterion='gini', max_depth=300, min_samples_split=3, min_samples_leaf=6, random_state=42	0.93	0.93	0.93	0.414	0.005

❖ **Kết quả huấn luyện mô hình Decision Tree theo nghi thức K-fold, K = 15**

Bảng 8: Bảng kết quả huấn luyện mô hình Decision Tree nghi thức 15-fold

		Precision	Recall	F1 score	Training time	Inference time
Decision Tree	criterion='gini', max_depth=100, min_samples_split=3, min_samples_leaf=6, random_state=42	0.934	0.934	0.934	0.406	0.002
	criterion='gini', max_depth=200, min_samples_split=3, min_samples_leaf=6, random_state=42	0.934	0.934	0.934	0.664	0.006
	criterion='gini', max_depth=300, min_samples_split=3, min_samples_leaf=6, random_state=42	0.934	0.934	0.934	0.547	0.003

4. Giải thuật Random Forest

4.1. Giới thiệu

Giải thuật Random Forest là một thuật toán học máy có giám sát, được sử dụng để phân loại và hồi quy dữ liệu. Thuật toán này xây dựng một tập hợp các cây quyết định (decision tree), sau đó sử dụng kết quả bỏ phiếu của các cây quyết định này để đưa ra dự đoán.

4.2. Cách hoạt động của giải thuật

Giải thuật Random Forest hoạt động như sau:

- Đầu tiên, thuật toán sẽ xây dựng một tập huấn luyện gồm các điểm dữ liệu đã được phân loại.
- Sau đó, thuật toán sẽ xây dựng một số cây quyết định, mỗi cây sẽ được xây dựng trên một tập con ngẫu nhiên của tập huấn luyện.
- Khi có một điểm dữ liệu mới cần được phân loại hoặc dự đoán, thuật toán sẽ sử dụng kết quả bỏ phiếu của các cây quyết định để đưa ra dự đoán.

4.3. Ưu điểm, nhược điểm của giải thuật

- **Ưu điểm:**
  - Độ chính xác cao.
  - Có khả năng chống overfitting.
  - Có thể áp dụng cho nhiều loại dữ liệu khác nhau.
- **Nhược điểm:**
  - Có thể bị chậm khi triển khai.
  - Có thể khó giải thích.

4.4. Kết quả huấn luyện mô hình

- **Kết quả huấn luyện mô hình Random Forest theo nghi thức Hold-out, datatrain 80% và datatest 20%**

Bảng 9: Bảng kết quả huấn luyện mô hình Random Forest nghi thức hold-out

		Precision	Recall	F1 score	Training time	Inference time
Random Forest	n_estimators=200, criterion='gini', max_depth=100, min_samples_split=3, min_samples_leaf=6, random_state=42	0.962	0.962	0.962	18.312	0.461
	n_estimators=300, criterion='gini', max_depth=100, min_samples_split=3, min_samples_leaf=6, random_state=42	0.962	0.961	0.961	26.738	0.618
	n_estimators=300, criterion='gini', max_depth=200, min_samples_split=3, min_samples_leaf=6, random_state=42	0.962	0.961	0.961	26.369	0.632
	n_estimators=200, criterion='gini', max_depth=200, min_samples_split=5, min_samples_leaf=10, random_state=42	0.954	0.954	0.954	17.281	0.403

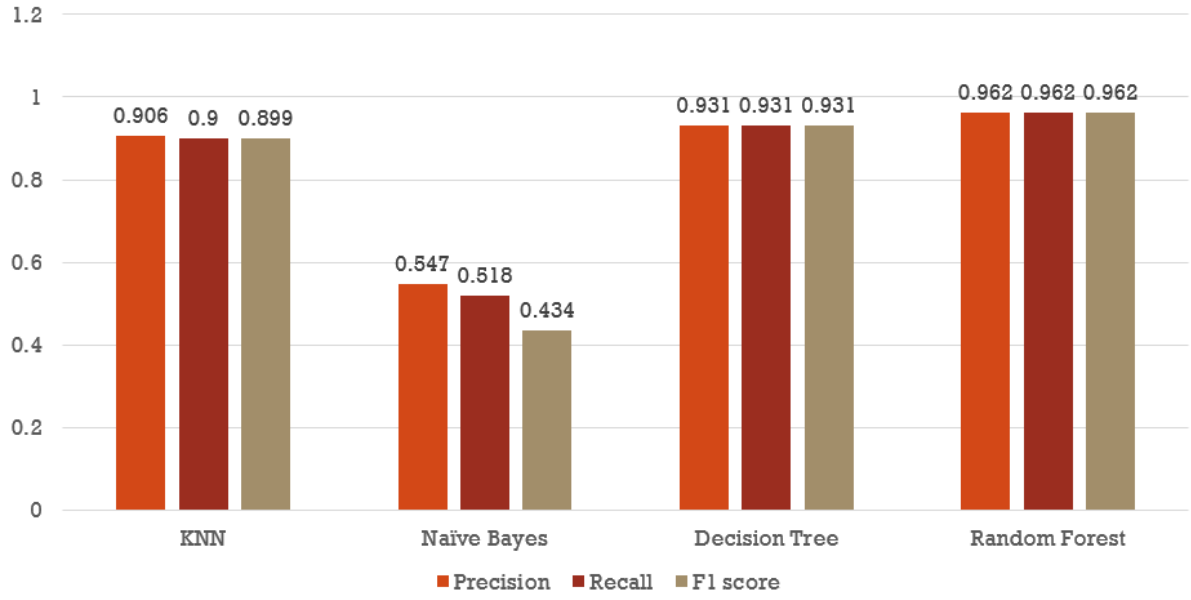
❖ Kết quả huấn luyện mô hình Random Forest theo nghi thức K-fold, K = 15

Bảng 10: Bảng kết quả huán luyện mô hình Random Forest nghi thức 15-fold

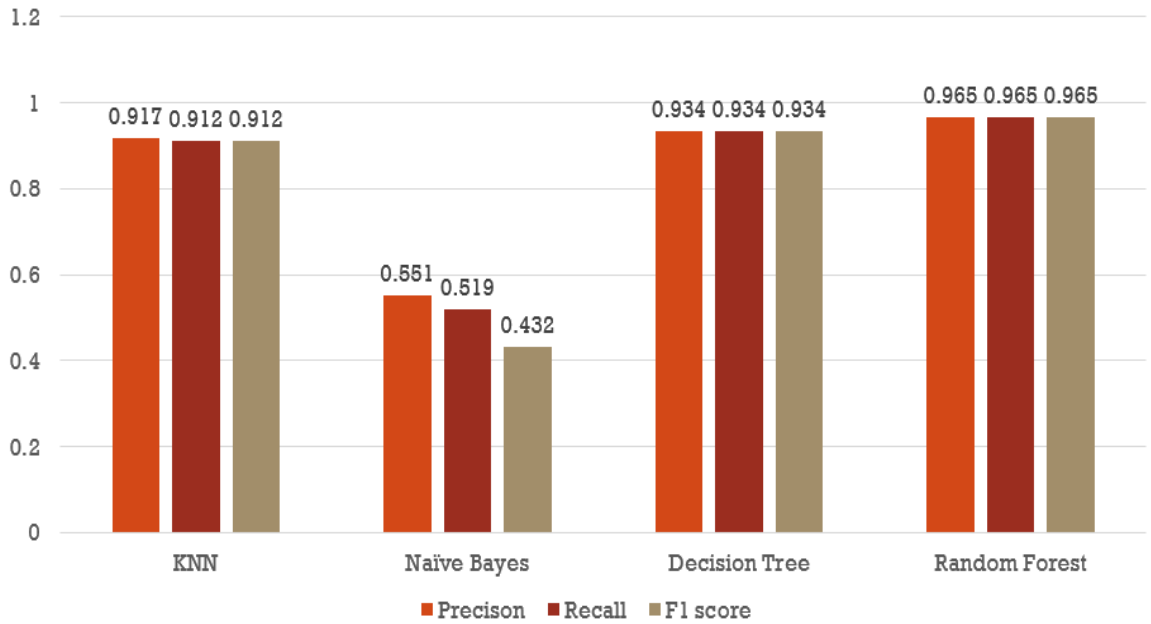
		Precision	Recall	F1 score	Training time	Inference time
Random Forest	n_estimators=200, criterion='gini', max_depth=100, min_samples_split=3, min_samples_leaf=6, random_state=42	0.965	0.965	0.965	20.054	0.155
	n_estimators=300, criterion='gini', max_depth=100, min_samples_split=3, min_samples_leaf=6, random_state=42	0.965	0.965	0.965	30.413	0.253
	n_estimators=300, criterion='gini', max_depth=200, min_samples_split=3, min_samples_leaf=6, random_state=42	0.965	0.965	0.965	30.409	0.246
	n_estimators=200, criterion='gini', max_depth=200, min_samples_split=5, min_samples_leaf=10, random_state=42	0.958	0.957	0.957	19.699	0.156

CHƯƠNG III. ĐÁNH GIÁ VÀ TRIỂN KHAI MÔ HÌNH

1. Đánh giá mô hình phân lớp



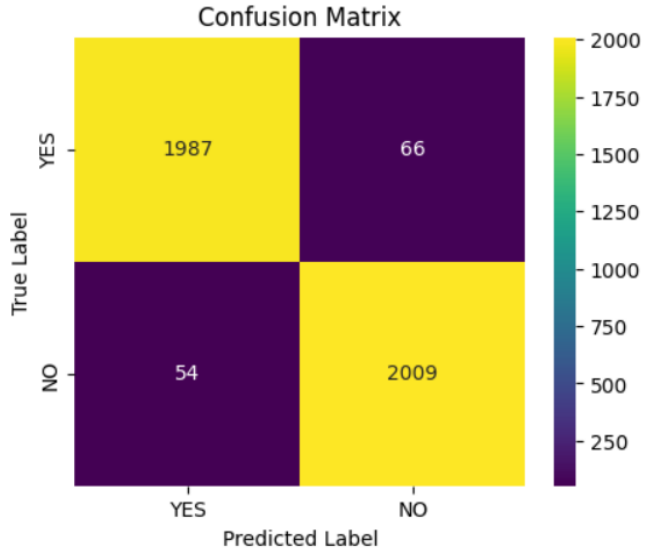
Hình 16: Kết quả độ chính xác của từng mô hình theo nghi thức đánh giá hold-out



Hình 17: Kết quả độ chính xác của từng mô hình theo nghi thức đánh giá 15-fold

Ở cả hai nghi thức đánh giá, mô hình Naïve Bayes đều cho ra kết quả độ chính xác thấp nhất, các mô hình còn lại: KNN, Decision Tree, Random Forest đều cho ra kết quả độ chính xác cao (> 89%). Mô hình Random Forest cho ra độ chính xác cao nhất: 96.2% (nghi thức hold-out) và 96.5% (nghi thức k-fold) => Lựa chọn nghi thức k-fold, mô hình Random Forest để triển khai mô hình thành ứng dụng.

2. Nhận xét kết quả thực nghiệm



Hình 18: Ma trận nhầm lẫn mô hình Random Forest

Mô hình ở hình 18 dự đoán ở lớp YES đúng được 1987/2053 mẫu. Mô hình dự đoán lớp NO đúng được 2009/2063 mẫu. Mô hình trên dự đoán đúng được tất cả 3996/4116 mẫu => Độ chính xác của mô hình trên là 97%.

3. Triển khai mô hình

Mô hình máy học được phát triển để giải quyết các vấn đề thực tế và cụ thể trong lĩnh vực nào đó. Bằng cách triển khai mô hình thành ứng dụng, chúng ta có thể áp dụng giải pháp vào môi trường thực tế để cải thiện hoặc tối ưu hóa các quy trình và quyết định. Khi một mô hình máy học được tích hợp vào một ứng dụng, người dùng cuối có thể trực tiếp tận hưởng lợi ích mà mô hình mang lại mà không cần phải hiểu rõ về cách nó hoạt động bên trong. Điều này làm cho công nghệ máy học trở nên dễ sử dụng hơn và có thể áp dụng rộng rãi.

Thư viện hỗ trợ triển khai mô hình:

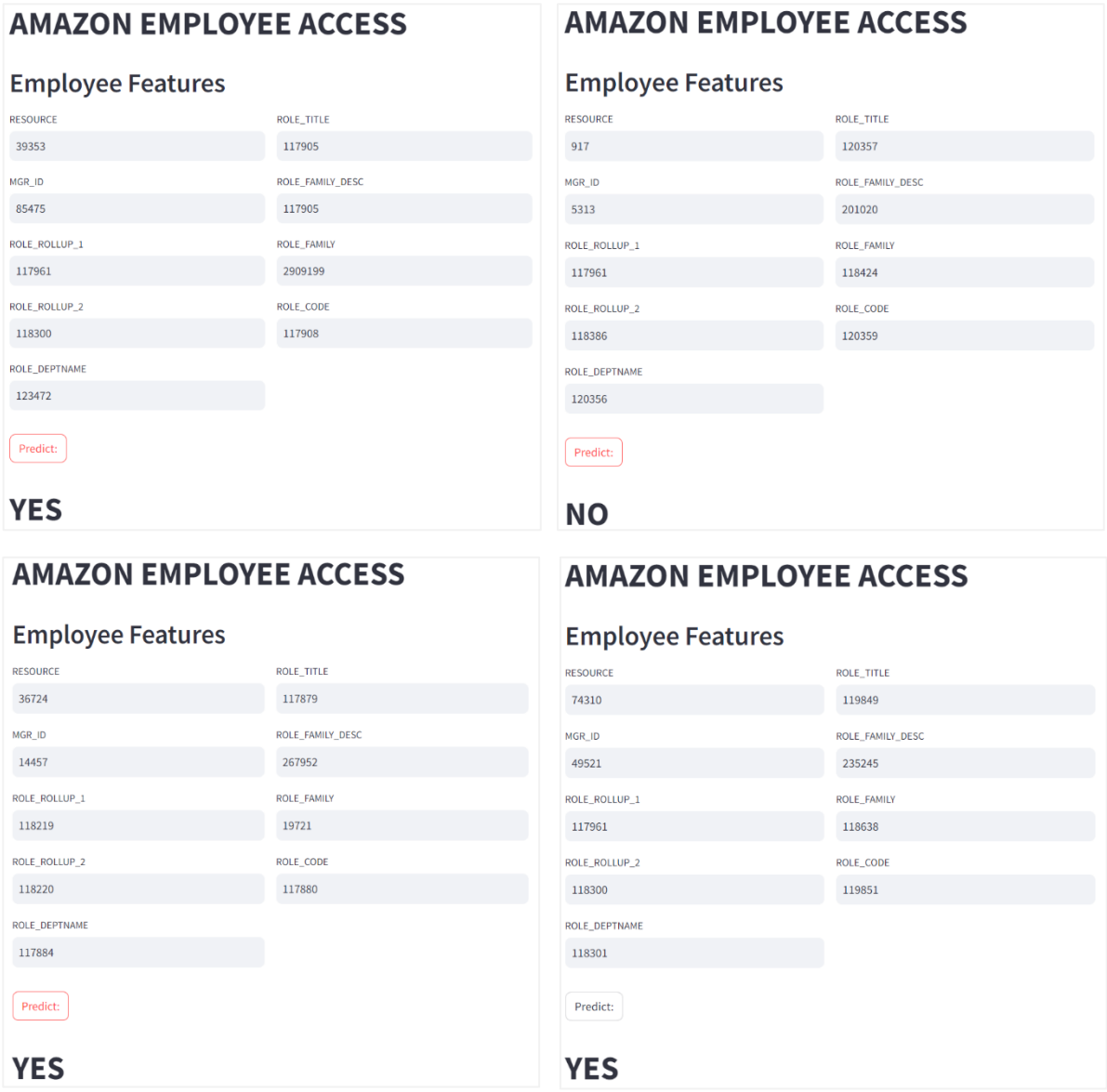
- **Thư viện streamlit:** Streamlit là một thư viện Python mã nguồn mở được sử dụng để tạo các ứng dụng web tương tác một cách nhanh chóng và dễ dàng. Thư viện này cung cấp một bộ công cụ GUI mạnh mẽ cho phép các nhà khoa học dữ liệu, nhà phân tích dữ liệu và nhà phát triển web tạo các ứng dụng web đẹp mắt và hiệu quả chỉ với một vài dòng mã.

- **Thư viện joblib:** Joblib là một thư viện Python mã nguồn mở được sử dụng để lưu trữ và phục hồi các đối tượng Python. Thư viện này dựa trên NumPy và SciPy, và nó cung cấp các phương pháp hiệu quả để lưu trữ các đối tượng Python dưới dạng các tệp nhị phân. Có thể sử dụng joblib để lưu mô hình máy học.

❖ Bảng test case kiểm thử ứng dụng:

Bảng 11: Bảng test case kiểm thử ứng dụng

Test case	**Dữ liệu kiểm thử**	**Kết quả mong đợi**	**Kết quả thực tế**	**Pass/ Fail**
TC01	resource=39353, mgr_id=85475, role_rollup_1=117961, role_rollup_2=118300, role_deptname=123472, role_title=117905, role_family_desc=117905, role_family=2909199, role_code=117908	YES	YES	Pass
TC02	resource=917, mgr_id=5313, role_rollup_1=117961, role_rollup_2=118386, role_deptname=120356, role_title=120357, role_family_desc=201020, role_family=118424, role_code=120359	NO	NO	Pass
TC03	resource=36724, mgr_id=14457, role_rollup_1=118219, role_rollup_2=118220, role_deptname=117884, role_title=117879, role_family_desc=267952, role_family=19721, role_code=117880	YES	YES	Pass
TC04	resource=74310, mgr_id=49521, role_rollup_1=117961, role_rollup_2=118300, role_deptname=118301, role_title=119849, role_family_desc=235245, role_family=118638, role_code=119851	NO	YES	Fail



Hình 19: Kết quả kiểm thử mô hình đã được triển khai

PHẦN KẾT LUẬN

1. Kết quả đạt được

- Xây dựng mô hình máy học với độ chính xác 96%.
- Triển khai thành công mô hình thành ứng dụng với sự hỗ trợ của thư viện Streamlit.

2. Hướng phát triển

- Triển khai mô hình chạy trên nền tảng website/ mobile app.
- Tích hợp mô hình máy học vào hệ thống nhúng.

TÀI LIỆU THAM KHẢO

[1]. Andreas C. Muller & Sarah Guido, Introduction to Machine Learning with Python, O'Reilly Media, 2016.

[2]. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[3]. Andrew Ng, Machine Learning Yearning, https://github.com/ajaymache/machine-learning-yearning