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

ĐỒ ÁN
HỌC PHẦN MÁY HỌC ỨNG DỤNG

Đề tài

XÂY DỰNG MÔ HÌNH DỰ ĐOÁN PHÊ DUYỆT TÍN DỤNG KHÁCH HÀNG

Nhóm sinh viên thực hiện:
1. Trương Văn Anh Kiệt B2205886
2. Phan Thành Tiến B2205814
3.Nguyễn Quàn Thắng B2205908

Cần Thơ, 07/2025

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

ĐỒ ÁN
HỌC PHẦN MÁY HỌC ỨNG DỤNG

Đề tài
XÂY DỰNG MÔ HÌNH DỰ ĐOÁN PHÊ DUYỆT TÍN DỤNG KHÁCH HÀNG

Giảng viên hướng dẫn:
Lưu Tiến Đạo

Nhóm sinh viên thực hiện:
1. Trương Văn Anh Kiệt B2205886
2. Phan Thành Tiến B2205814
3.Nguyễn Quàn Thắng B2205908

Cần Thơ, 07/2025

NHẬN XÉT CỦA GIẢNG VIÊN

-

-

-

-

-

-

-

Cần Thơ, ngày tháng năm

(Ký và ghi rõ họ tên)

MỤC LỤC

DANH MỤC HÌNH **5**
PHÂN CÔNG CÔNG VIỆC **6**
PHẦN NỘI DUNG **7**
1. Mô tả bài toán : **7**
1.1. Tóm tắt: 7
1.2. Mục đích: 7
2. Mô tả dữ liệu, ý nghĩa của dữ liệu : **8**
3. Phân tích dữ liệu và lựa chọn mô hình **9**
3.1. Phân tích dữ liệu: 9
3.2. Lựa chọn mô hình: 14
3.2.1. Cơ sở lí thuyết KNN: 14
3.2.2. Cơ sở lí thuyết Bayes: 15
3.2.3. Cơ sở lí thuyết Decision Tree: 16
4. Cấu hình máy tính **17**
5. Huấn luyện mô hình **17**
5.1. Huấn luyện mô hình KNN: 17
5.2. Huấn luyện mô hình Decision tree: 18
5.3. Huấn luyện mô hình Bayes: 20
6. Đánh giá mô hình: **21**
6.1. Đánh giá bằng độ chính xác trung bình : 21
6.2. Đánh giá bằng ma trận nhầm lẫn: 22
6.2.1. Ma trận nhầm lẫn KNN: 22
6.2.2. Ma trận nhầm lẫn Decision Tree: 23
6.2.3. Ma trận nhầm lẫn Bayes: 24
6.3. Đánh giá bằng các chỉ số Precision / Recall / F1-score: 25
PHẦN KẾT LUẬN **27**
1. Kết quả đạt được: **27**
2. Hướng phát triển : **27**
TÀI LIỆU THAM KHẢO **28**

DANH MỤC HÌNH

Hình 1 .Bộ dữ liệu Credit Approval Dataset 8

Hình 2. Biểu đồ phân bố kết quả phê duyệt tín dụng 14

Hình 3. Biểu đồ thể hiện độ chính xác của mô hình qua 10 lần chạy 18

Hình 4. Biểu đồ thể hiện độ chính xác của mô hình Decision Tree 19

Hình 5. Mô hình cây quyết định bậc 4 19

Hình 6. Biểu đồ thể hiện độ chính xác của mô hình Bayes 20

Hình 7. Kết quả đánh giá độ chính xác của các mô hình qua 10 lần huấn luyện 21

Hình 8. Ma trận nhầm lẫn KNN 22

Hình 9. Ma trận nhầm lẫn Decision Tree 23

Hình 10. Ma trận nhầm lẫn Bayes 24

Hình 11. Biểu đồ đánh giá mở rộng mô hình Decision Tree 25

PHÂN CÔNG CÔNG VIỆC

Họ tên và MSSV	Nhiệm vụ	Đánh giá hoàn thành công việc
Trương Văn Anh Kiệt B2205886	Hỗ trợ hoàn thành đồ án, đưa ra ý kiến đóng góp và chỉnh sửa đồ án, viết báo cáo đồ án	100%
Phan Thành Tiến B2205914	Hỗ trợ hoàn thành đồ án, tìm kiếm thông tin về thuật toán và đưa ra ý kiến đóng góp	100%
Nguyễn Quàn Thắng B2205908	Thiết kế slide powerpoint, hỗ trợ hoàn thành đồ án. Đưa ra ý kiến đóng góp và chỉnh sửa	100%

PHẦN NỘI DUNG

1. Mô tả bài toán :

1.1. Tóm tắt:

Tín dụng là một hình thức cấp vốn hoặc cho vay mà trong đó một bên (nguoiiwf cho vay) cung cấp tiền, hàng hóa hoặc dịch vụ cho bên người đi vay với cam kết rằng trong một khoảng thời gian được định sẳn khoản vay sẽ được hoàn trả đúng hẹn, bên canh đó sẽ có một phần lãi suất nhất định

Phê duyệt tín dụng là quá trình một tổ chức tài chính như ngân hàng hay công ty thẻ tín dụng đánh giá và quyết định xem có nên chấp nhận cho một các nhân hay tổ chức vay tín dụng hay không dựa trên một số đặc điểm như tuổi tác, trình độ học vấn, tình trạng việc làm ,thu nhập hàng tháng,….Những yếu tố và đặc điểm trên góp phần giúp cho công ty tài chính hay ngân hàng phân loại khách hàng và xem xét xem có nên phê duyệt tín dụng cho khách hàng hay không. Trong bài báo cáo này, nhóm chúng tôi nghiên cứu và phân loại khách hàng theo phương pháp học máy có giám sát. Các thuật toán được dùng là KNN,Decision tree (Cây quyết định) và Naive Bayes. Đây là những thuật toán cơ bản được áp dụng rộng rãi ở nhiều lĩnh vực trong cuộc sống như y tế, tài chính, quản lí khách hàng, nhận dạng hình ảnh,…..

1.2. Mục đích:

Mục đích của dự án là xác định đặc điểm, yếu tố nào trong tập dữ liệu là quan trọng nhất để xác định xem đơn xin thẻ tín dụng có nên được chấp thuận hay từ chối hay không và liệu mô hình mấy học có thể dự đoán chính xác kết quả của đơn xin hay không.

Tuy nhiên, việc lựa chọn hay tùy chỉnh các tham số chỉ mang tính tương đối và có thể bị ảnh hưởng bởi nhiều yếu tố khác. Trong phần báo cáo này, nhóm chúng tôi sử dụng tập dữ liệu Credit Approval Dataset trên UCI Machine Learning Repository với tổng số mẫu là 690 với 15 thuộc tính khác nhau hứa hẹn sẽ giúp việc tìm ra những phân khúc khách hàng có thể được chấp nhận vay tín dụng từ nó nâng cao hiệu suất làm việc cho các công ty tài chính.

2. Mô tả dữ liệu, ý nghĩa của dữ liệu :

Bộ dữ liệu Credit Approval Dataset (Hình 1)đã thu thập được tổng cộng 690 điểm dữ liệu từ khách hàng với 15 thuộc tính khác nhau từ A1 đến A15 và nhãn là + (phê duyệt) và - (không phê duyệt).

U8

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16
2	b	30.83	0	u	g	w	v	1.25	t	t	01	f	g	00202	0	+
3	a	58.67	4.46	u	g	q	h	3.04	t	t	06	f	g	00043	560	+
4	a	24.5	0.5	u	g	q	h	1.5	t	f	0	f	g	00280	824	+
5	b	27.83	1.54	u	g	w	v	3.75	t	t	05	t	g	00100	3	+
6	b	20.17	5.625	u	g	w	v	1.71	t	f	0	f	s	00120	0	+
7	b	32.08	4	u	g	m	v	2.5	t	f	0	t	g	00360	0	+
8	b	33.17	1.04	u	g	r	h	6.5	t	f	0	t	g	00164	31285	+
9	a	22.92	11.585	u	g	cc	v	0.04	t	f	0	f	g	00080	1349	+
10	b	54.42	0.5	y	p	k	h	3.96	t	f	0	f	g	00180	314	+
11	b	42.5	4.915	y	p	w	v	3.165	t	f	0	t	g	00052	1442	+
12	b	22.08	0.83	u	g	c	h	2.165	f	f	0	t	g	00128	0	+
13	b	29.92	1.835	u	g	c	h	4.335	t	f	0	f	g	00260	200	+
14	a	38.25	6	u	g	k	v	1	t	f	0	t	g	00000	0	+
15	b	48.08	6.04	u	g	k	v	0.04	f	f	0	f	g	00000	2690	+
16	a	45.83	10.5	u	g	q	v	5	t	t	07	t	g	00000	0	+
17	b	36.67	4.415	y	p	k	v	0.25	t	t	10	t	g	00320	0	+
18	b	28.25	0.875	u	g	m	v	0.96	t	t	03	t	g	00396	0	+
19	a	23.25	5.875	u	g	q	v	3.17	t	t	10	f	g	00120	245	+
20	b	21.83	0.25	u	g	d	h	0.665	t	f	0	t	g	00000	0	+
21	a	19.17	8.585	u	g	cc	h	0.75	t	t	07	f	g	00096	0	+
22	b	25	11.25	u	g	c	v	2.5	t	t	17	f	g	00200	1208	+
23	b	23.25	1	u	g	c	v	0.835	t	f	0	f	s	00300	0	+
24	a	47.75	8	u	g	c	v	7.875	t	t	06	t	g	00000	1260	+
25	a	27.42	14.5	u	g	x	h	3.085	t	t	01	f	g	00120	11	+
26	a	41.17	6.5	u	g	q	v	0.5	t	t	03	t	g	00145	0	+

Hình 1 .Bộ dữ liệu Credit Approval Dataset

Như hình 1, chúng ta nhận thấy rằng các thuộc tính đã bị ẩn và được thay thế từ A1 đến A15. Các thuộc tính từ A1 đến A15 bao gồm cả các thuộc tính định tính và định lượng. Ví dụ như A1 chỉ có các gia tri là a,b còn A2 là biểu thị một con số cụ thể. Vì thế, để tiện cho việc phân loại và đánh giá, nhóm chúng tôi tiến hành gán nhãn giả định cho từng thuộc tính trong bảng sau:

A1	Giới tính
A2	Tuổi
A3	Mức nợ hiện tại
A4	Tình trạng học vấn
A5	Là khách hàng ngân hàng (hay không)
A6	Trình độ học vấn
A7	Tình trạng nhà ở
A8	Số năm đã làm việc
A9	Có từng vỡ nợ
A10	Tình trạng việc làm
A11	Diểm tín dụng
A12	Có bằng lái xe
A13	Tình trạng quốc tế
A14	Mã vùng
A15	Thu nhập

3. Phân tích dữ liệu và lựa chọn mô hình

3.1. Phân tích dữ liệu:

Bộ dữ liệu khá đa dạng gồm nhiều loại dữ liệu khác nhau chứa cả dữ liệu liên tục và rời rạc và có một số dữ liệu bị thiếu nằm trong 7 thuộc tính khác nhau, có khoảng hơn 30 bản ghi bị thiếu. Dữ liệu cũng khá cân bằng, trường hợp chấp thuận

chiếm 307/690 (44,5%) và từ chối chiếm 383/690 (55,5%). Ngoài ra dữ liệu của một số thuộc tính định lượng có giá trị liên tục và có phạm vi chênh lệch khá lớn nên cần được chuẩn hóa để đảm bảo không phụ thuộc quá nhiều vào 1 thuộc tính nào.

```
# Chuẩn hóa dữ liệu
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

for name, model in models.items():
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)
    acc = accuracy_score(y_test, y_pred)
    results[name].append(acc)
```

Một số thuật toán học máy phổ biến như KNN, Decision Tree,SVM,… không thể xử lý dữ liệu bị thiếu vì thế nếu không xử lý truecj tiếp các gia tri bị thiếu, một số mô hình học máy có thể không hoạt động hoặc bị lỗi. Mặc khác, giá trị bị thiếu nếu không xử lý có thể làm lệch kết quả dự đoán. Chính vì thế, kiểm tra dữ liệu trước khi training là cực kỳ cần thiết. Ta có thể kiểm tra dữ liệu bị thiếu bằng hàm df.isnull().sum() trong tập pandas

```
Kiem tra du lieu bị thieu:
A1     12
A2     12
A3      0
A4      6
A5      6
A6      9
A7      9
A8      0
A9      0
A10     0
A11     0
A12     0
A13     0
A14    13
A15     0
A16     0
dtype: int64
```

=> Dữ liệu trong tập Credit Approval đang bị thiếu ở dòng A1,A2,A4,A5,A6,A7,A14 => dữ liệu bị thiếu khá nhiều

Hiện tại, có nhiều thuật toán chuyên dùng để xử lý dữ liệu bị thiếu, ưu điểm của các thuật toán này là hạn chế khá nhiều việc mất các thông tin quan trọng như simple imputer, KNN imputer, Iterative pmputer,….Nhưng trong báo cáo này, để đơn giản hóa nhóm chúng tôi thực hiện thay thế thủ công. Nếu là dữ liệu dạng chữ, tìm kiếm giá trị xuất hiện nhiều nhất trong cột dữ liệu để điền vào giá trị bị thiếu. Nếu giá trị dữ liệu là dạng số, tiến hành tính giá trị trung bình của cột dữ liệu để điền vào.

```python
# Xử lý dữ liệu bị thiếu
for col in df.columns:
    if df[col].dtype == 'object':
        df[col] = df[col].fillna(df[col].mode()[0])
    else:
        df[col] = df[col].fillna(df[col].mean())
```

Quan sát hình 1, ta thấy rằng tập dữ liệu Credit Approval có các cột có giá trị là dạng kiểu chữ. Tuy nhiên, máy tính thì không thể hiểu hay xử lí được các giá trị có dạng kiểu chữ hay chuỗi mà chỉ có thể xử lí các gia trị dạng số. Vì thế, trước khi tiến hành các giải thuật máy học, ta cần chuyển đổi các dữ liệu có dạng chữ (chuỗi) có trong tập dữ liệu Credit Approval sang dạng số. Có nhiều cách chuyển đổi dữ liệu sang kiểu số hiện nay, trong báo cáo này, nhóm chúng tôi sử dụng thuật toán tự động mã hóa toàn bộ các cột dạng chữ bằng One-Hot-Encoding

=> Vậy One-Hot-Encoding là gì?

One-Hot-Encodinglà một kỹ thuật trong tiền xử lý dữ liệu, được sử dụng để chuyển đổi các giá trị dạng chữ thành các số nguyên để máy tính có thể xử lý được gọn gàng.

Khác với việc gán số nguyên trực tiếp cho các giá trị (như 0, 1, 2, 3...), One-Hot Encoding sẽ tạo ra một cột nhị phân (0 hoặc 1) cho mỗi giá trị duy nhất trong một cột. Nếu một cột có n giá trị khác nhau, sẽ sinh ra n cột mới.

Như vậy, One-Hot Encoding giúp tránh được hiểu lầm về khoảng cách giữa các giá trị như khi dùng Label Encoding (gán a=0, b=1, c=2,...), vốn có thể ảnh hưởng đến các thuật toán học máy nhạy với khoảng cách như KNN hoặc SVM.

Hiểu đơn gián rằng, nếu ta gán giá trị thủ công là 0,1,2,3 thì khoảng cách từ giá trị mới tới các giá trị có trong cột dữ liệu sẽ khác nhau gây nhầm lẫn cho các mô hình học máy dựa vào khoảng cách nhưng nếu ta gán giá trị nhị phân cho từng giá trị ví dụ cột đó có 4 giá trị là a,b,c,d thì One-Hot-Encoding sẽ mã hóa thành 4 cột nhị phân như sau:

	a	b	c	d
a	1	0	0	0
b	0	1	0	0
c	0	0	1	0
d	0	0	0	1

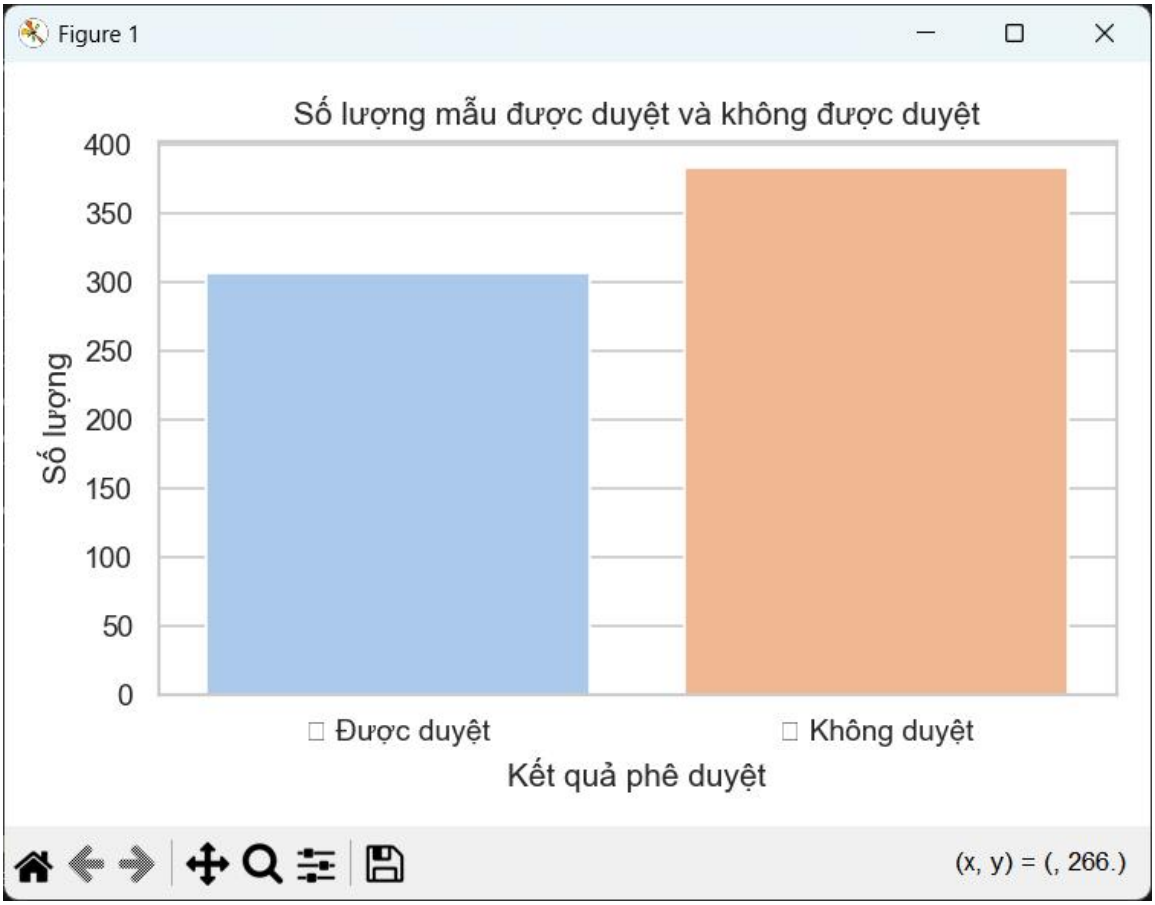
=> Như vậy, các thuật toán sẽ coi 4 giá trị này là như nhau thay vì xem 1<2<3<4 như Label Encoding. Tuy nhiên đối với nhãn (A16) ta bắt buộc dùng Label Encoding thay vì dùng One-Hot-Encoding bởi vì giá trị nhãn là duy nhất không nên tách nhãn ra làm 2 cột có thể ảnh hưởng đến kết quả dự đoán.

Vì thế, trong báo cáo này, nhóm chúng tôi tiến hành mã hóa bằng One-Hot-Encoding trong tất cả cột dữ liệu trừ cột nhãn(A16) và mã hóa cột nhãn bằng Label-Encoding.

```
# Encode nhãn A16
le_y = LabelEncoder()
df['A16'] = le_y.fit_transform(df['A16'])  # '+' → 1, '-' → 0
label_mapping = dict(zip(le_y.classes_, le_y.transform(le_y.classes_)))
print("📌 Ánh xạ nhãn A16:", label_mapping)

# One-Hot Encoding các cột object (trừ A16)
df = pd.get_dummies(df, columns=df.select_dtypes(include=['object']).columns, drop_first=True)

# Tách X và y
X = df.drop(columns=['A16'])
y = df['A16']
```



Hình 2. Biểu đồ phân bố kết quả phê duyệt tín dụng

3.2. Lựa chọn mô hình:

Trong báo cáo này, nhóm chúng tôi lựa chọn 3 thuật toán máy học là KNN, Decision Tree và Naive Bayes vì một vài lí do sao:

+ Bayes khá phù hợp với dữ liệu bị thiếu, dữ liệu có nhiều đặc trưng rời rạc và đơn giản có tính nhanh và ổn định

+ Decision Trê có thể mô hình hóa quan hệ giữa các thuộc tính và kết quả cho kết quả trực quan lý do tại sao phê duyệt tại sao không

+ KNN sau khi làm việc với dữ liệu đã chuẩn hóa sẽ cho kết quả khá chính xác, không yêu cầu cao về dữ liệu

3.2.1. Cơ sở lí thuyết KNN:

Thuật toán K-Nearest Neighbors (KNN) là một trong những thuật toán học máy đơn giản và phổ biến nhất thuộc nhóm học có giám sát (supervised learning). KNN được sử dụng chủ yếu cho các bài toán phân loại (classification) và hồi quy (regression), tuy nhiên phổ biến hơn là trong phân loại.

KNN dự đoán các phần tử mới dựa trên số lượng thông tin "phổ biến" nhất của k láng giềng gần nhất với nó. KNN còn có tên gọi khác là Lazy-learning

Để xác định được lớp phần tử mới cần:

- Tính toán khoảng cách từ phần tử mới đến các phần tử còn lại trong tập huấn luyện
- Chọn k phần tử gần nhất với phần tử trong tập huấn luyện
- Gán nhãn cho phần tử mới bằng nhãn " phổ biến" nhất của k láng giềng gần nhất

Khoảng cách có thể tính theo kiểu số, kiểu rời rạc, kiểu nhị phân tùy theo datasets

❖ Ưu điểm:

- Dễ cài đặt, dễ hiểu, đơn giản
- Không có quá trình học, không có mô hình xây dựng
- Hiệu quả với dữ liệu có biên độ rõ ràng giữa các lớp

- Không cần giả định về phân phối dữ liệu
- Có thể làm việc với nhiều loại dữ liệu

❖ Nhược điểm:

- Tính toán chậm khi dữ liệu qua lớn vì cần tính khoảng cách từ phần tử mới tới tất cả các điểm
- Nhạy cảm với dữ liệu nhiễu và không đồng đều
- Độ phức tạp của quá trình phân loại lớn
- Kết quả phụ thuộc vào khoảng cách

❖ Lưu ý:

- Không chọn k quá lớn hoặc quá nhỏ
- Nên chọn k là số lẻ để tránh hòa phiếu

3.2.2. Cơ sở lí thuyết Bayes:

Thuật toán Naive Bayes là một mô hình phân loại thuộc nhóm học có giám sát (supervised learning), dựa trên định lý Bayes trong xác suất thống kê. Đây là một phương pháp đơn giản nhưng rất hiệu quả, đặc biệt trong các bài toán phân loại văn bản, phân loại thư rác (spam), và các hệ thống gợi ý.

Với Bayes, ta phải xây dựng và huấn luyện mô hình để tính xác suất xuất hiện của tất cả các trường hợp . Khi dữ liệu mới cần dự đoán được đưa vào, ta cần phân loại và xác định nhãn của đối tượng mới thông qua giá trị xác xuất lớn nhất tính được

❖ Ưu điểm:

- Dễ cài đặt, học nhanh, kết quả trực quan dễ hiểu
- Rất nhanh, hiệu quả với dữ liệu lớn do đã tính toán xác xuất từ ban đầu
- Ít yêu cầu tài nguyên tính toán
- Hoạt động tốt ngay cả khi giả định độc lập không hoàn toán đúng
- Có thể xử lý dữ liệu bị nhiễu

❖ Nhược điểm:

- Không thể học tương tác giữa các đặc trưng
- Giả định độc lập giữa các thuộc tính không hoàn toàn đúng có thể ảnh hưởng đến độ chính xác
- Không hiệu quả khi có quá nhiều thuộc tính dư thừa

3.2.3. Cơ sở lí thuyết Decision Tree:

Cây quyết định là một mô hình học máy thuộc nhóm học có giám sát (supervised learning), được sử dụng cho cả phân loại (classification) và hồi quy (regression). Mô hình này hoạt động bằng cách chia nhỏ dữ liệu thành các nhóm nhỏ hơn theo dạng cây phân nhánh, dựa trên các thuộc tính (đặc trưng) của dữ liệu.

Cây quyết định là giải thuật học:

- Kết quả sinh ra dễ diễn dịch
- Khá đơn giản, nhanh, hiệu quả được sử dụng nhiều
- Liên tục trong nhiều năm qua, cây quyết định được bình chọn là giải thuật được sử dụng nhiều nhất và thành công nhất.
- Giải quyết các vấn đề của phân loại và hồi quy
- Làm việc cho dữ liệu số và kiểu liệt kê

Một cây quyết định bao gồm:

- Nút gốc : bắt đầu quá trình phân tích
- Nút trong đại diện cho các điều kiện kiểm tra trên đặc trưng
- Cành(branches): biểu diễn kết quả của phép kiểm tra
- Nút lá: đưa ra dự đoán nhãn (kết quả của phân loại/hồi quy)

Quá trình học cây là tìm cách phân chia dữ liệu ở từng bước sao cho tăng mức độ "thuần khiết" của từng nhóm – tức là nhóm lại các điểm có nhãn giống nhau

Tại mỗi nút, các thuộc tính được đánh giá dựa trên phân tách dữ liệu học tốt nhất có thể. Việc đánh giá tốt hay xấu dựa trên các heuristics: độ lợi thông tin, tỉ số độ lợi thông tin, chỉ số gini

❖ Ưu điểm:

- Dễ hiểu và trực quan
- Không cần chuẩn hóa dữ liệu
- Hoạt động tốt với dữ liệu có mối quan hệ phi tuyến

❖ Nhược điểm:

- Dễ học vẹt nếu cây quá sâu
- Nhạy cảm với dữ liệu nhiễu
- Cây có thể không ổn định nếu dữ liệu thay đổi nhỏ

4. Cấu hình máy tính

❖ Phần cứng:

- CPU: Từ Intel i5 trở lên.
- RAM: Từ 8GB trở lên.
- Ổ cứng: tối thiểu 1GB dung lượng trống

❖ Phần mềm:

- Python.
- Các thư viện bổ trợ (Pandas, Scikit-learn, ...)

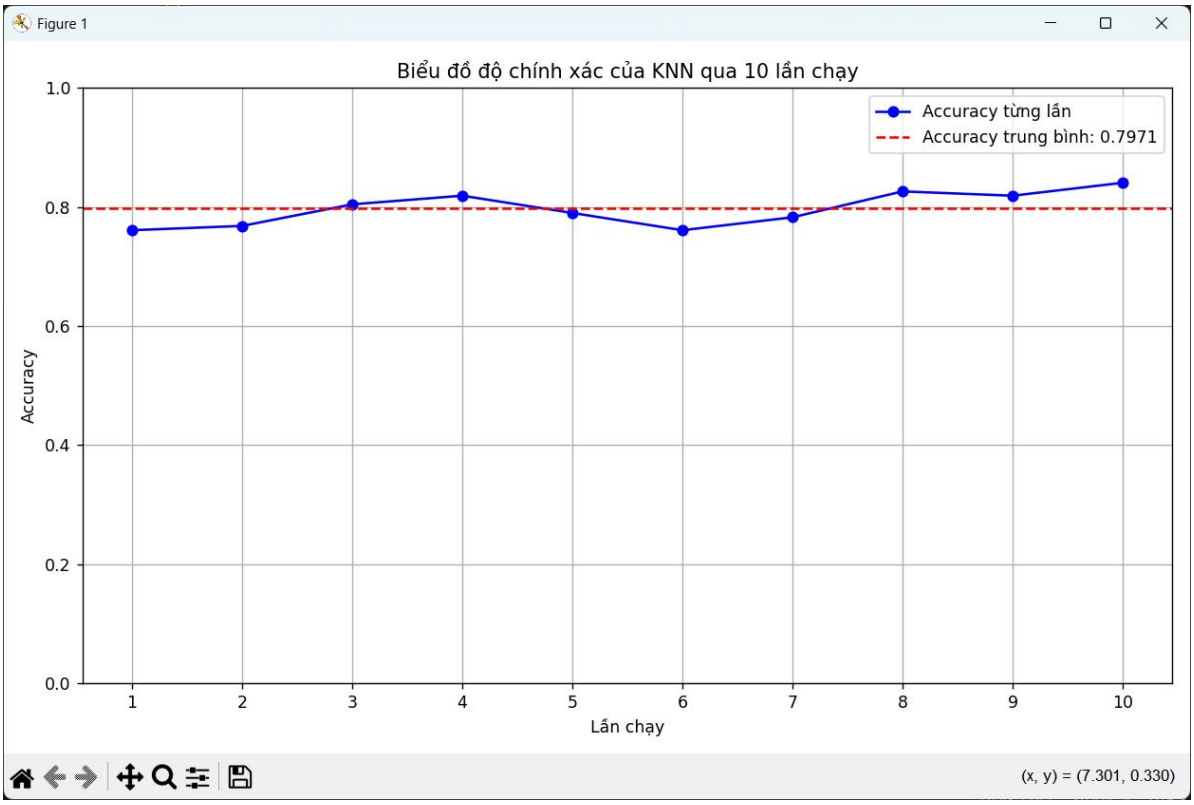
5. Huấn luyện mô hình

5.1. Huấn luyện mô hình KNN:

```
model = KNeighborsClassifier(n_neighbors=5)
results = []
```

=> Huấn luyên KNN với k=5

Tiến hành chạy giải thuật 10 lần



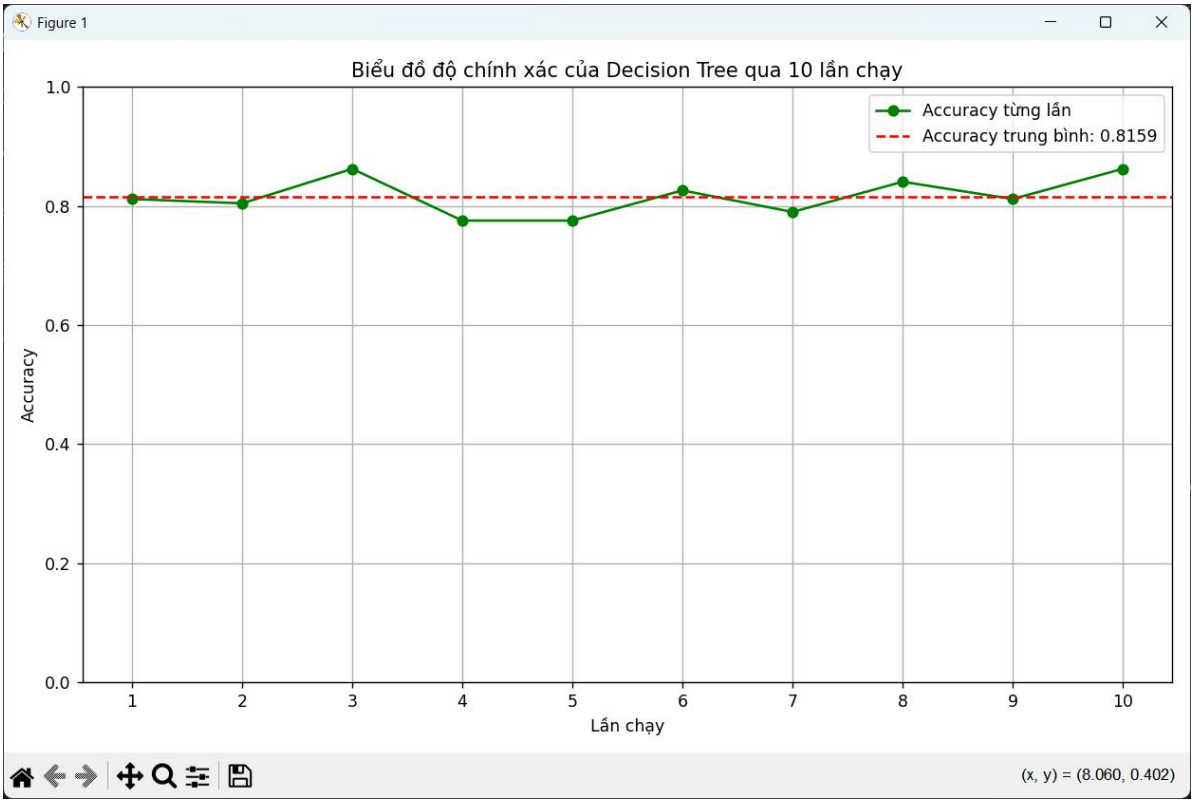
Hình 3. Biểu đồ thể hiện độ chính xác của mô hình qua 10 lần chạy

5.2. Huấn luyện mô hình Decision tree:

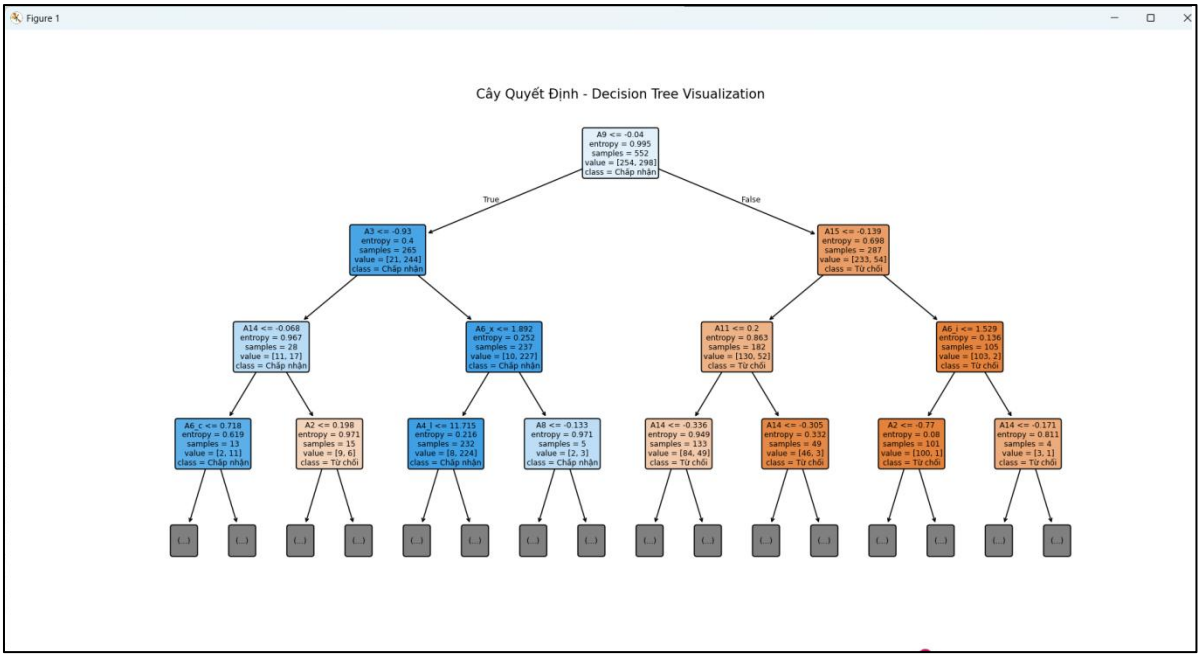
```
model = DecisionTreeClassifier(criterion='entropy', random_state=0)
results = []
```

=> Huấn luyện mô hình Cây quyết định với chỉ số entropy để phân chia dữ liệu

Tiến hành chạy giải thuật 10 lần



Hình 4. Biểu đồ thể hiện độ chính xác của mô hình Decision Tree



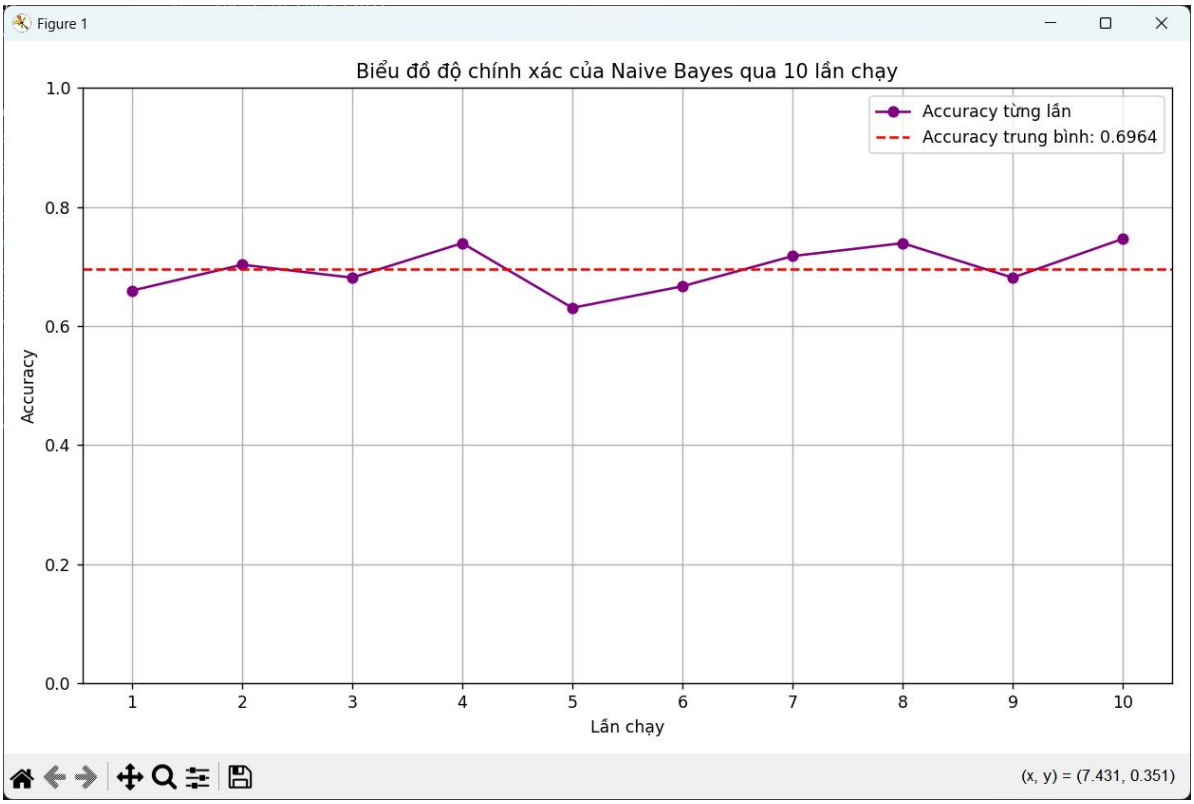
Hình 5. Mô hình cây quyết định bậc 4

Tiến hành vẽ cây quyết định (cố định ở bậc 4 để nhỏ gọn, trực quan) của tập dữ liệu và ta được cây như hình 4

5.3. Huấn luyện mô hình Bayes:

```
model = GaussianNB()
results = []
```

Tiến hành chạy giải thuật 10 lần:



Hình 6. Biểu đồ thể hiện độ chính xác của mô hình Bayes

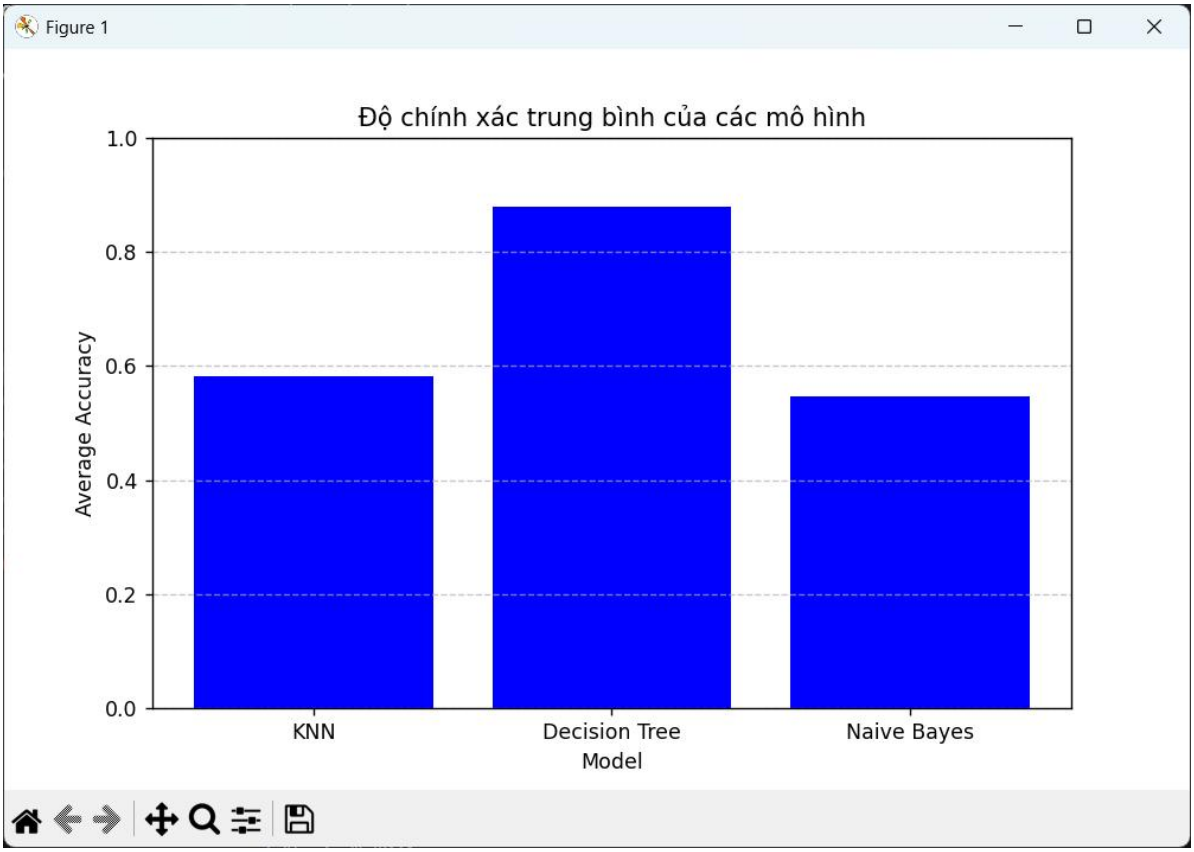
❖ Đánh giá tổng quan:

- Quan sát Hình 4, hình 5, hình 6; ta dễ dàng nhận thấy độ chính xác qua từng lần chạy và độ chính xác trung bình của Decision Tree là cao nhất, nằm ở ngưỡng khoảng 80% . Lí do có thể là dữ liệu của tập dữ liệu phân tách rõ ràng, có những quy tắc cố định khiến cho Decision Tree học khá tốt vì Decision Tree phân tách đặc trưng theo các điều kiện cụ thể, đồng thời do tập dữ liệu bao gồm cả biến liên tục và phân loại mà Cây quyết định thì làm việc với chúng khá tốt. Ngoài ra, Cây quyết định còn làm khá tốt với các thuộc tính có mối quan hệ phi tuyến hay kết hợp nhiều đặc trưng lại với nhau.
- Về Bayes, ta thấy độ chính xác qua từng lần chạy và độ chính xác trung bình khá thấp nằm ở ngưỡng từ 60 đến 70%. Lí do có thể là Bayes giả định các

thuộc tính độc lập với nhau nhưng tuy nhiên, trong thực tế, các thuộc tính trong tập dữ liệu Credit Approval có mối liên hệ mật thiết với nhau ở nhiều khía cạnh. Do giả định ban đầu là sai nên mô hình dự đoán dễ bị sai lệch và không chính xác. Mặc khác, Bayes thiên về tính xác suất nên không thể học hay hiểu được các mối quan hệ phi tuyến hay kết hợp nhiều đặc trưng

6. Đánh giá mô hình:

6.1. Đánh giá bằng độ chính xác trung bình :

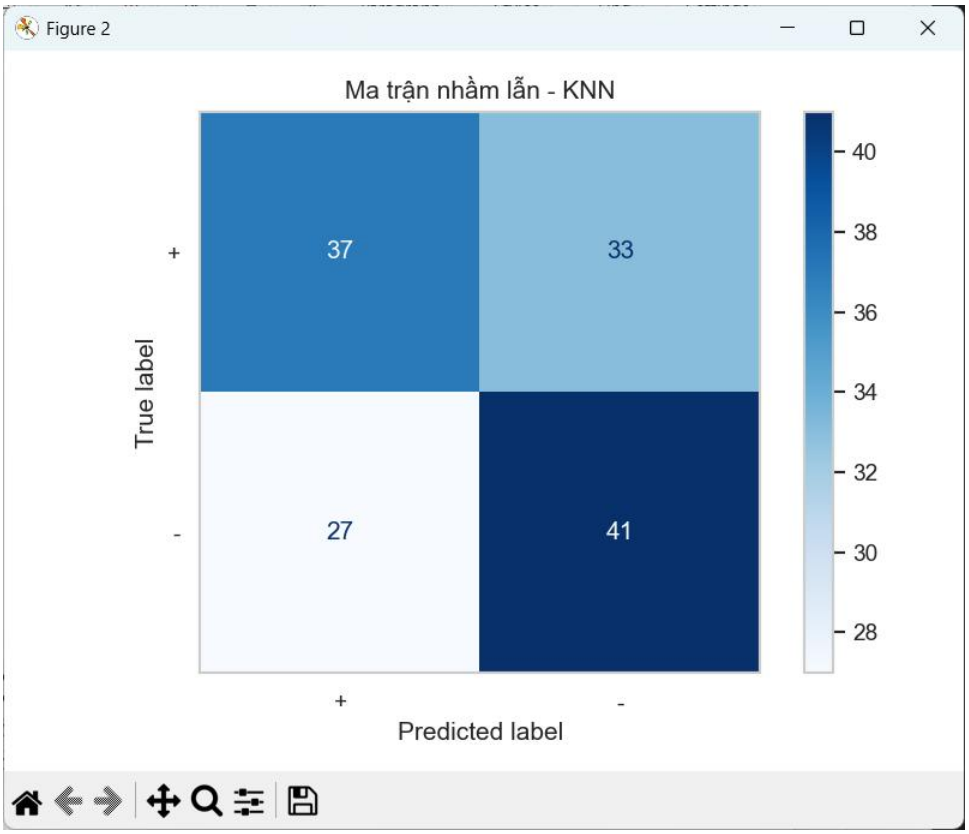


Hình 7. Kết quả đánh giá độ chính xác của các mô hình qua 10 lần huấn luyện

=> Độ chính xác trung bình của Decision Tree là vượt trội so với KNN và Bayes chứng tỏ tập dữ liệu Credit Approval rất phù hợp sử dụng Decision Tree

6.2. Đánh giá bằng ma trận nhầm lẫn:

6.2.1. Ma trận nhầm lẫn KNN:

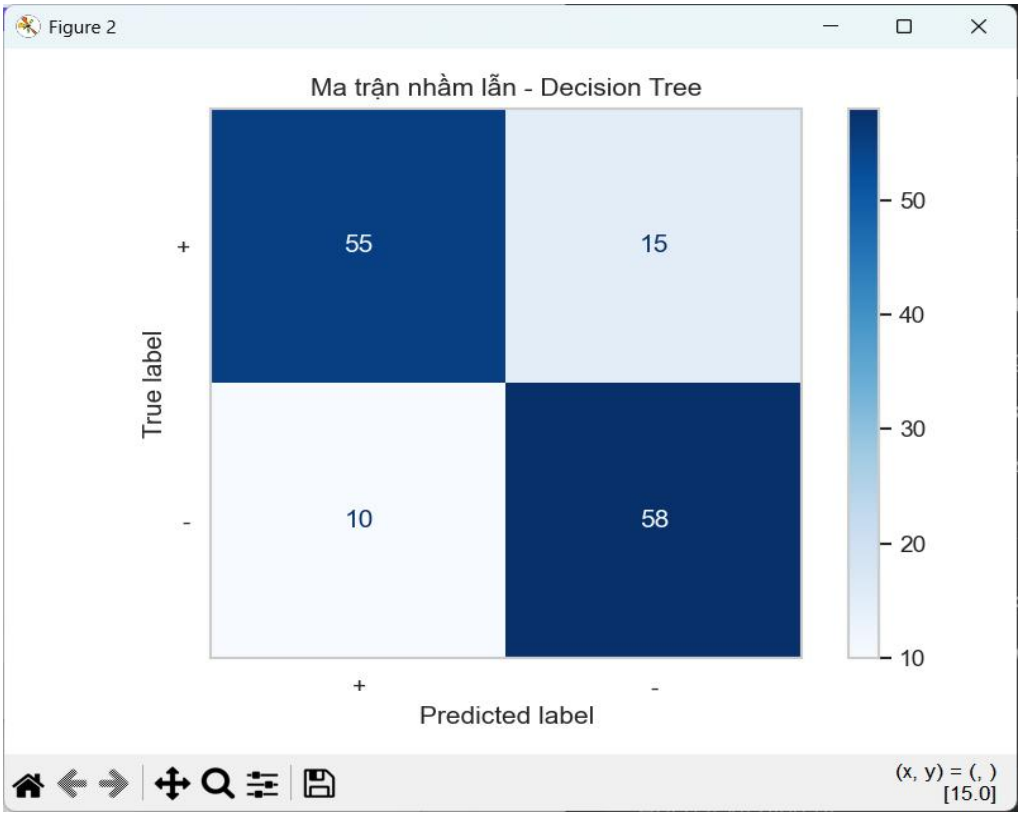


Hình 8. Ma trận nhầm lẫn KNN

❖ Nhận xét:

- Trong số 70 thuộc tính mang nhãn là + (được duyệt) thì mô hình KNN dự đoán chỉ đúng 37/70 tức chỉ đúng khoảng 52.85% . Như vậy, mô hình đã từ chối nhầm gần như một nửa số lượng khách hàng đáng ra phải được duyệt => độ chính xác khá thấp và thiếu sự tin cậy
- Trong số 68 thuộc tính mang nhãn là - (Không duyệt) thì mô hình KNN dự đoán đúng 41/68 tức khoảng 60.29%. Như vậy, mô hình đã chấp nhận nhầm khoảng 40% khách hàng đáng ra phải bị từ chối=> độ chính xác ở mức chấp nhận được, chưa phải quá cao
- Với tổng số lượng mẫu là 138, mô hình chỉ dự đoán đúng 37 + 41 = 78 mẫu, tức độ chính xác chỉ khoảng 56.5% => độ chính xác chỉ ở mức trung bình, tạm chấp nhận cần có phương hướng để nâng cao độ chính xác hoặc đề xuất thuật toán khác tin cậy hơn.

6.2.2. Ma trận nhầm lẫn Decision Tree:

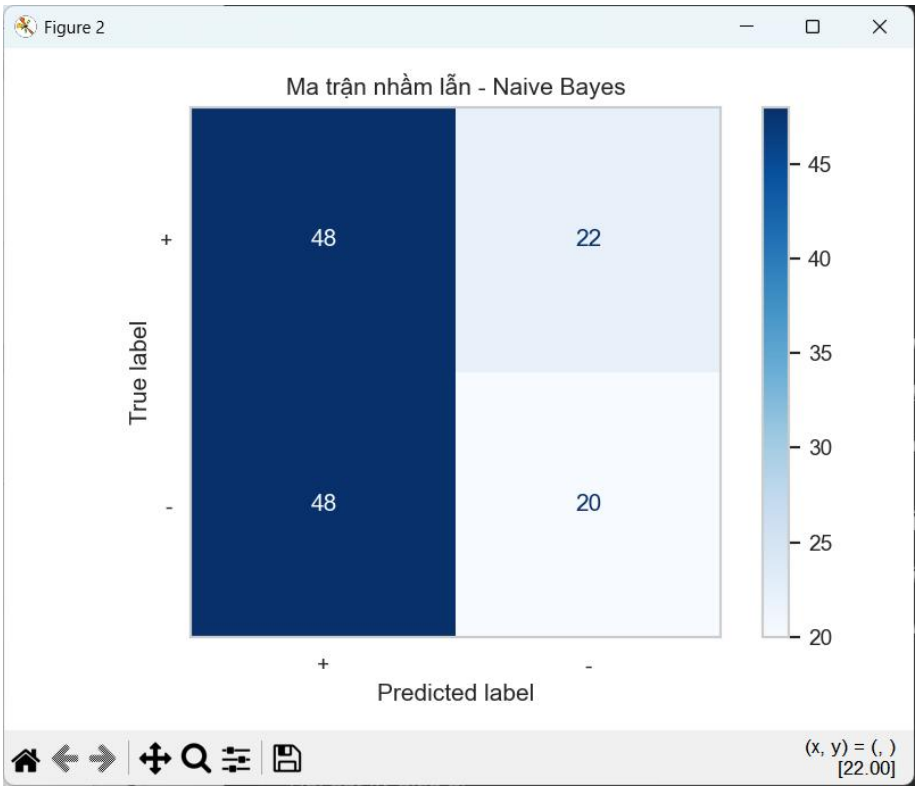


Hình 9. Ma trận nhầm lẫn Decision Tree

❖ Nhận xét:

- Trong số 70 thuộc tính mang nhãn là + (được duyệt) thì mô hình Decision Tree đã dự đoán đúng tới 55/70 tức độ chính xác nằm ở mức 78.57% . Như vậy, mô hình chỉ từ chối nhầm 15/70 khách hàng, một con số có thể chấp nhận được cao hơn nhiều so với KNN. Tuy nhiên, cũng cần có phương án cải tiến để giảm thiểu những trường hợp từ chối nhầm hết mức có thể.
- Trong số 68 thuộc tính mang nhãn là - (không duyệt) thì mô hình từ chối đúng 58/68 tức 85.29%. Độ chính xác ở mức khá cao vượt trội hơn hẳn KNN => độ tin cậy và độ chính xác cao, cần ưu tiên phát triển và nâng cấp
- Độ chính xác tổng thể đạt 81.88%

6.2.3. Ma trận nhầm lẫn Bayes:



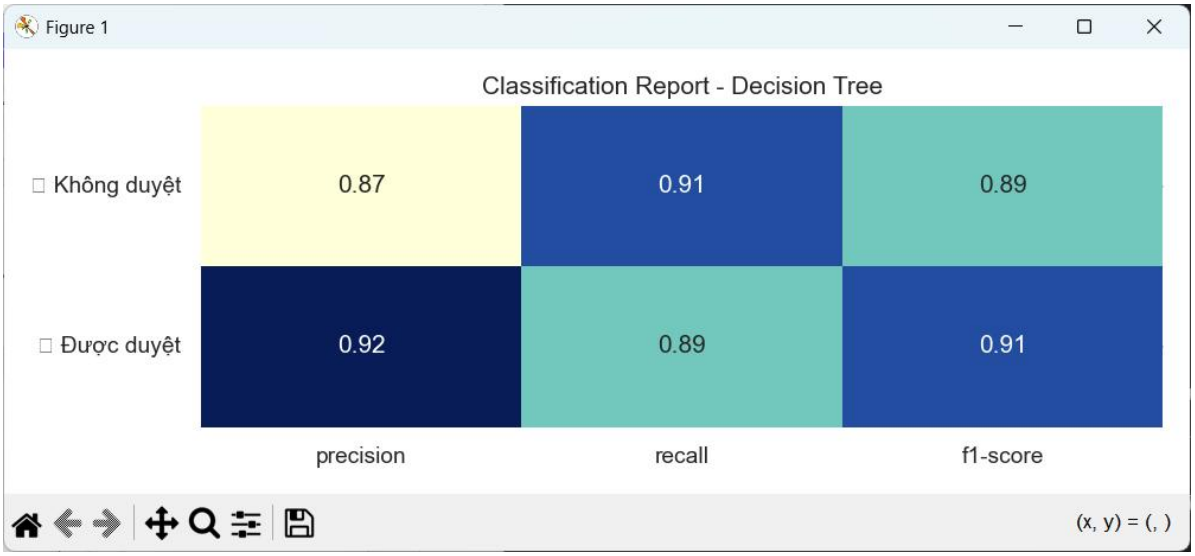
Hình 10. Ma trận nhầm lẫn Bayes

❖ Nhận xét:

- Tương tự như 2 mô hình trên, với nhãn + (được duyệt) đạt 68.57% và nhãn - (không duyệt) đạt 29.41% . Khác với KNN và Decision Tree, Bayes là mô hình có độ lệch giữa nhãn + và nhãn - là cao nhất cho thấy Bayes có sự thiên vị hơn với nhãn + hơn là nhãn -
- Ta không chỉ thấy rõ điều đó qua độ chính xác mà còn có thể quan sát trên hình 10, quan sát ta thấy trong tổng số 138 mẫu thì Bayes đã chấp nhận 96 mẫu chiếm gần 70% tổng số mẫu và độ chính xác trung bình đạt 49.28% => độ chính xác khá thấp=> không đáng tin cậy=> cần có phương án cải thiện hoặc thay thế

6.3. Đánh giá bằng các chỉ số Precision / Recall / F1-score:

Qua ma trận nhầm lẫn ta thấy được rằng mô hình Decision Tree là mô hình tốt nhất và đáng tin cậy nhất hiện tại. Vì thế, chúng tôi tiến hành đánh giá Decision Tree bằng những thước đo hiệu suất khác được mở rộng từ ma trận nhầm lẫn là Precision (độ chính xác khi dự đoán dương tính), recall (Độ bao phủ) và F1-score(Độ trung hòa). Do được mở rộng từ ma trận nhầm lẫn nên nhóm chúng tôi chỉ tiến hành đánh giá thuật toán tốt và chính xác nhất, tạm bỏ qua KNN và Bayes.



Hình 11. Biểu đồ đánh giá mở rộng mô hình Decision Tree

❖ Nhận xét:

- Đối với lớp **Không duyệt**, trong số tất cả các trường hợp ta thấy tỉ lệ dự đoán là không duyệt đạt 87% => nằm ở mức cao. Trong số các trường hợp không duyệt, mô hình đã dự đoán đúng 91%, chỉ bỏ sót 9%=> tỉ lệ bỏ sót khá thấp. F1 đạt 89% cho thấy sự cân bằng khá tốt giữa precision và recall => Mô hình đáng tin cậy khi từ chối đúng khách hàng không nên duyệt.
- Đối với lớp **Được duyệt**, trong số tất cả các trường hợp ta thấy rằng tỉ lệ dự đoán khách hàng được duyệt đạt ngưỡng 92%=> Độ chính xác rất cao => rất đáng tin cậy. Trong số các trường hợp được duyệt, mô hình đã chấp nhận đúng 89%=> tỉ lệ bỏ sót cũng rất thấp=> Chấp nhận đúng khách hàng tiềm năng

- Tóm tại, mô hình Decision Tree có độ chính xác và hoàn thiện cao, có sự cân bằng giữa được duyệt và không duyệt, không thiên lệch về một phía như Bayes=> Ưu tiên phát triển và nâng cấp

PHẦN KẾT LUẬN

1. Kết quả đạt được:

- Biết cách sử dụng ngôn ngữ lập trình Python để huấn luyện và đánh giá một mô hình học máy đơn giản
- Biết cách sử dụng các phương pháp tiền xử lý dữ liệu để xử lý dữ liệu hỗn loạn hay dữ liệu bị thiếu trước khi bắt đầu huấn luyện
- Biết cách sử dụng các thư viện có sẵn của python cho việc huấn luyện mô hình và vẽ biểu đồ
- Nắm bắt những điều cơ bản trong quá trình huấn luyện một mô hình học máy
- Hiểu được cơ sở lý thuyết của KNN, Decision Tree, Bayes và áp dụng chúng vào trong quá trình huấn luyện
- Biết cách lưu mô hình và đưa phần dự đoán mô hình lên web

2. Hướng phát triển :

- Cần cải thiện tốc độ và hiệu suất khi làm việc với dữ liệu lớn
- Ưu tiên cải tiến và phát triển những thuật toán có độ chính xác và tin cậy cao
- Đề xuất thay thế hoặc cải tiến các mô hình có độ chính xác và tin cậy thấp
- Thử nghiệm thêm các thuật toán tiến tiến và nâng cao (SVM, Random Forest)
- Phát triển giao diện người dùng triển khai trên web hoặc app để áp dụng vào môi trường thực tiển

TÀI LIỆU THAM KHẢO

[1]. Giáo trình Máy Học Ứng Dụng (Applied Machine Learning)

[2]. Chat GPT https://chatgpt.com/

[3] GitHub https://github.com/