Dự đoán phê duyệt khoản vay

Nguyễn Bá Khánh

Input

Thông tin khách hàng bao gồm: thu nhập, học vấn, khoản vay,...

Output

Khoản vay có được phê duyệt hay không?

Bộ dữ liệu

- Gồm 614 hàng
- 1 cột index Loan_ID
- 11 cột thông tin khoản vay
- 1 cột kết quả: Loan_Status

Mô tả sơ bộ dữ liệu

1. Có missing data:

Loan_ID	0
Gender	13
Married	3
Dependents	15
Education	0
Self_Employed	32
ApplicantIncome	0
CoapplicantIncome	0
LoanAmount	22
Loan_Amount_Term	14
Credit_History	50
Property_Area	0
Loan_Status	0

2. Dữ liệu thiếu cân bằng:

Tỷ lệ hồ sơ được duyệt vay chiếm ~69%

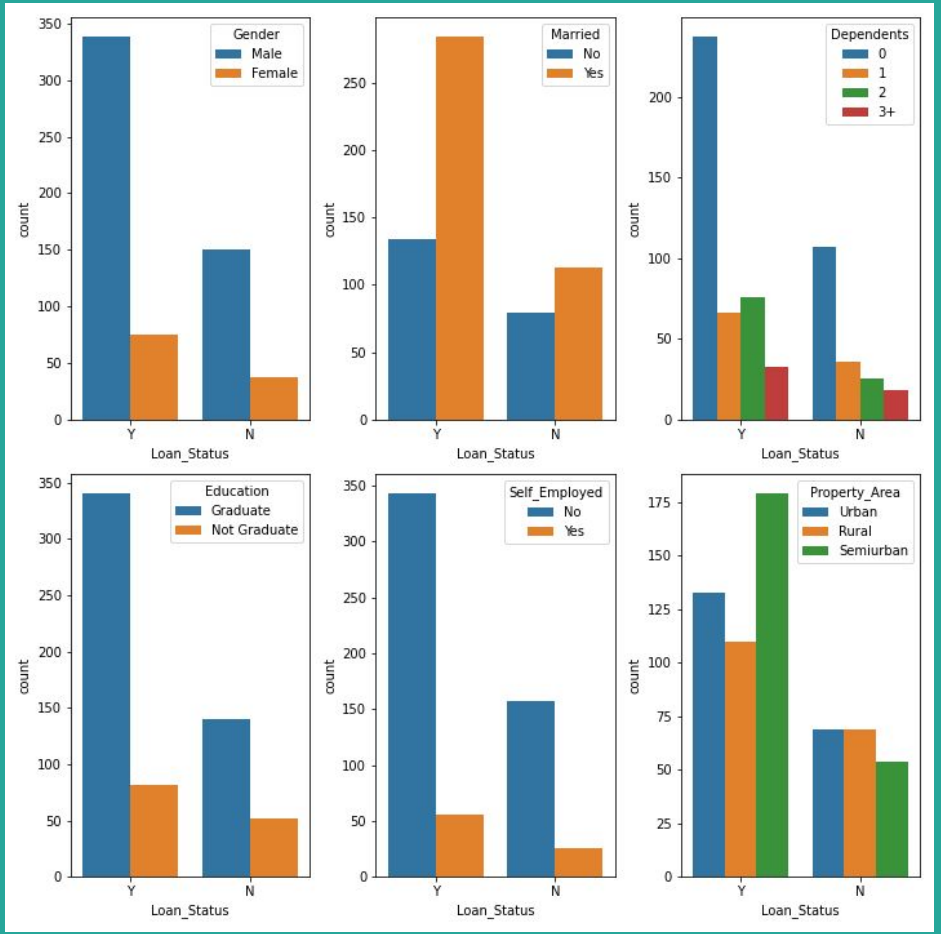
3. Các cột chưa được đưa về dạng số:

- Giới tính
- Tình trạng hôn nhân
- Người phụ thuộc
- Học vấn
- Tự doanh
- Khu vực tài sản bảo đảm

I. Xử lý dữ liệu

1. Loại bỏ cột Loan_ID vì chỉ đóng vai trò index
2. Kiểm tra các cột dữ liệu chưa được đưa về dạng số (kiểu dữ liệu object):
    - Giới tính: Male / Female / nan
    - Tình trạng hôn nhân: No / Yes / nan
    - Người phụ thuộc: 0 / 1 / 2 / 3+ / nan
    - Học vấn: Graduate / Not graduate
    - Tự doanh: No / Yes / nan
    - Khu vực tài sản bảo đảm: Urban / Rural / Semi-urban
    - Phê duyệt khoản vay: Y / N
3. Đưa cột phê duyệt khoản vay về dạng số: Y = 1, N = 0

4. Dùng biểu đồ để xem tương quan giữa các features và kết quả



Nhận thấy các features không có nhiều ảnh hưởng lắm tới kết quả, do tỷ lệ tăng ở mỗi feature gần tương đương với tỷ lệ tăng của data mẫu

=> Áp dụng onehot để đổi về dạng số và fillna bằng mode.

Riêng cột người phụ thuộc, nhận thấy số lượng người phụ thuộc từ 1 trở lên không có nhiều khác biệt, chủ yếu phân loại theo có và không có người phụ thuộc

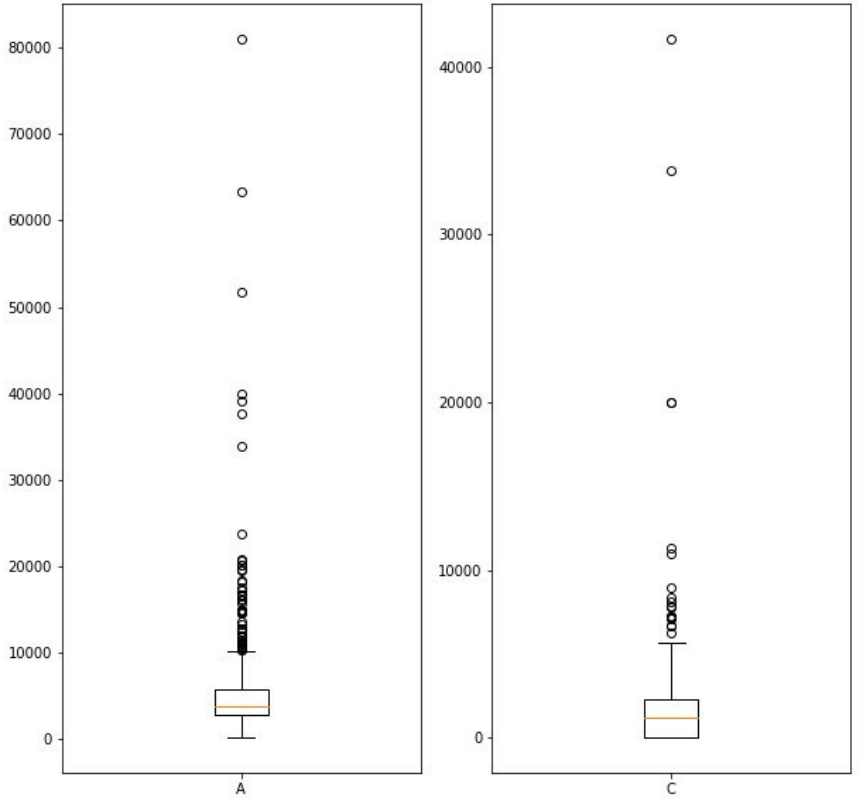
=> Áp dụng onehot để đưa về dạng số.

Cột khu vực tài sản có thể ảnh hưởng tới kết quả, giữ lại cả 3 cái và đưa về dạng số: 1, 10, 100.

Kiểm tra lại dataset:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	1	0	0	1	0	5849	0.0	NaN	360.0	1.0	1	1
1	1	1	1	1	0	4583	1508.0	128.0	360.0	1.0	100	0
2	1	1	0	1	1	3000	0.0	66.0	360.0	1.0	1	1
3	1	1	0	0	0	2583	2358.0	120.0	360.0	1.0	1	1
4	1	0	0	1	0	6000	0.0	141.0	360.0	1.0	1	1

5. Xử lý outlier ở cột thu nhập người vay và thu nhập người cùng vay



Xem các outliers ở cột thu nhập người vay (applicantincome > 30000)

Hầu hết đều được duyệt do thu nhập lớn => khả năng duyệt vay cũng cao hơn.

=> Hàng số 409 có thể do dữ liệu bị sai => loại bỏ khỏi dataset

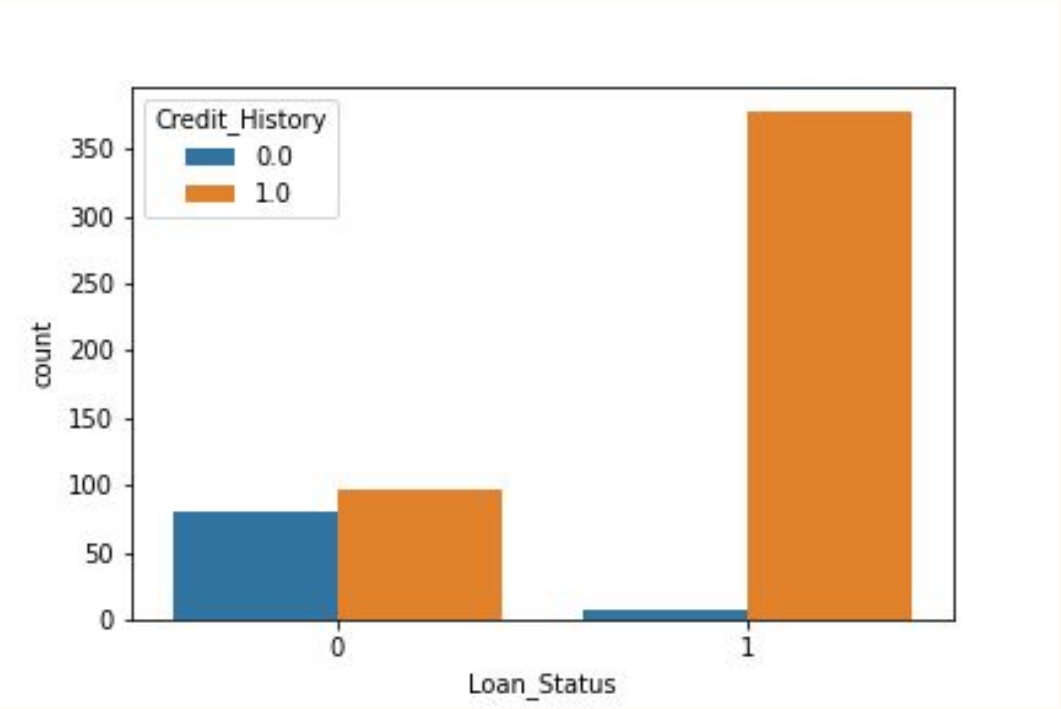
	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
155	1	1	1	1	0	39999	0.0	600.0	180.0	0.0	10	1
171	0	1	1	1	0	51763	0.0	700.0	300.0	1.0	1	1
183	1	1	1	1	0	33846	0.0	260.0	360.0	1.0	10	0
185	1	1	0	1	1	39147	4750.0	120.0	360.0	1.0	10	1
333	1	1	0	1	1	63337	0.0	490.0	180.0	1.0	1	1
409	1	1	1	1	0	81000	0.0	360.0	360.0	0.0	100	0
443	1	0	1	1	0	37719	0.0	152.0	360.0	1.0	10	1

Xem các outliers ở cột thu nhập người cùng vay (coappincome > 10000) và thử so sánh với các trường hợp có thu nhập người cùng vay trong khoảng từ 5k đến 10k.
=> Không nhận thấy quy luật đặc biệt, biến đổi hết về mean.

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
9	1	1	1	1	0	12841	10968.0	349.0	360.0	1.0	10	0
177	1	1	1	1	0	5516	11300.0	495.0	360.0	0.0	10	0
402	1	0	0	1	0	2500	20000.0	103.0	360.0	1.0	10	1
417	1	1	1	1	1	1600	20000.0	239.0	360.0	1.0	1	0
581	1	0	0	1	0	1836	33837.0	90.0	360.0	1.0	1	0
600	0	0	1	1	1	416	41667.0	350.0	180.0	NaN	1	0

6. Xử lý dữ liệu bị thiếu ở các cột còn lại

- Fillna ở 2 cột Khoản vay và Thời hạn vay bằng mean.
- Qua việc so sánh các dữ liệu ở trên, nhận thấy hồ sơ có Lịch sử tín dụng hầu hết đều được duyệt vay, vẽ đồ thị để kiểm tra:



Nhận định có vẻ chính xác, vì cột Lịch sử tín dụng có missing value nên nếu bỏ các data sẽ cho dataset khách quan hơn, tuy nhiên vì số lượng mẫu tương đối nhỏ, biến đổi về mean (tức là 1 - có lịch sử tín dụng)

7. Kiểm tra lại dataset

```
data_pre2.info()
```

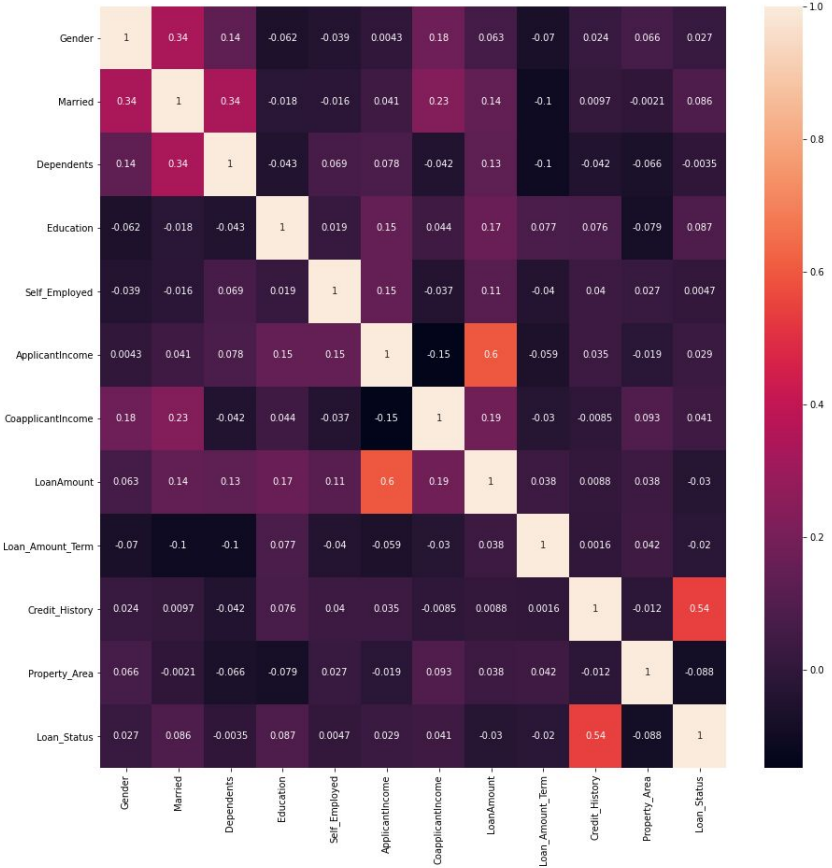
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 613 entries, 0 to 613
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Gender             613 non-null    int64  
 1   Married            613 non-null    int64  
 2   Dependents         613 non-null    int64  
 3   Education          613 non-null    int64  
 4   Self_Employed      613 non-null    int64  
 5   ApplicantIncome    613 non-null    int64  
 6   CoapplicantIncome  613 non-null    float64
 7   LoanAmount         613 non-null    float64
 8   Loan_Amount_Term   613 non-null    float64
 9   Credit_History     613 non-null    float64
 10  Property_Area      613 non-null    int64  
 11  Loan_Status        613 non-null    int64  
dtypes: float64(4), int64(8)
memory usage: 82.3 KB
```

8. Kiểm tra các cột dữ liệu quan trọng:

a. Xem bảng correlation

Nhận thấy có 4 cột dữ liệu ảnh hưởng nhiều nhất tới kết quả bao gồm:

- Lịch sử tín dụng
- Khu vực tài sản
- Học vấn
- Tình trạng hôn nhân



b. Dùng SelectKBest:

Nhận thấy có 5 cột dữ liệu ảnh hưởng nhiều nhất là:

- Thu nhập người vay
- Thu nhập người cùng vay
- Khoản vay
- Lịch sử tín dụng
- Khu vực tài sản

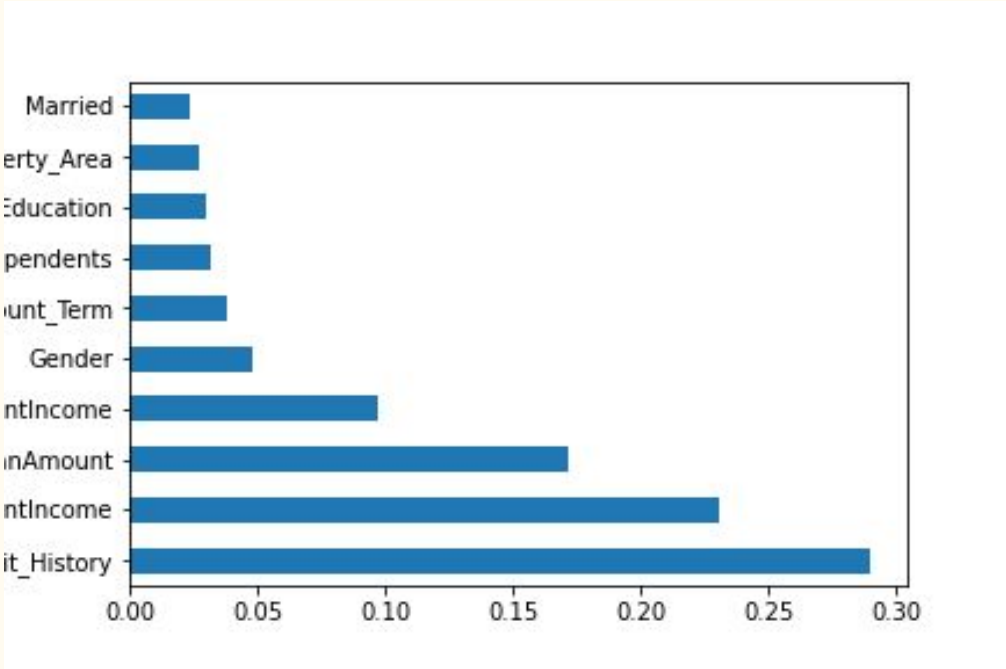
```
best_features = SelectKBest(score_func= chi2, k= 10)
features = best_features.fit(Xf, yf)
fscores = pd.DataFrame(features.scores_)
fcolumns = pd.DataFrame(Xf.columns)
topfeatures = pd.concat([fcolumns, fscores], axis= 1)
topfeatures.columns = ['Features', 'Score']
print(topfeatures)
# nhận thấy có 5 features ảnh hưởng lớn tới kết quả: Ap
```

	Features	Score
0	Gender	0.088998
1	Married	1.607027
2	Dependents	0.004278
3	Education	1.022150
4	Self_Employed	0.011075
5	ApplicantIncome	2770.705317
6	CoapplicantIncome	2019.700197
7	LoanAmount	27.134407
8	Loan_Amount_Term	3.070315
9	Credit_History	25.493732
10	Property_Area	264.440755

c. Dùng DecisionTree:

Nhận thấy có 5 cột dữ liệu ảnh hưởng nhiều nhất là:

- Lịch sử tín dụng
- Thu nhập người vay
- Khoản vay
- Thu nhập người cùng vay
- Giới tính



Chọn 5 cột dữ liệu:

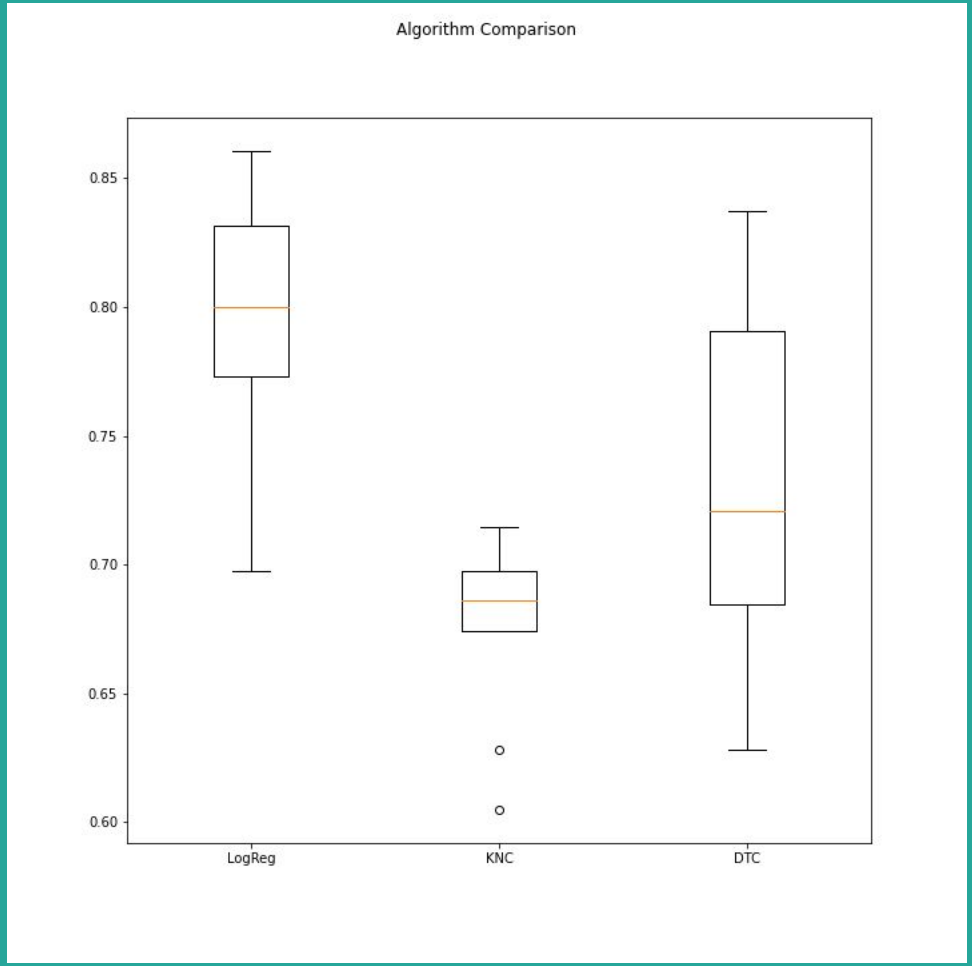
- Lịch sử tín dụng
- Thu nhập người vay
- Thu nhập người cùng vay
- Khoản vay
- Khu vực tài sản

Do được xác định là dữ liệu quan trọng ở 2/3 lần kiểm tra

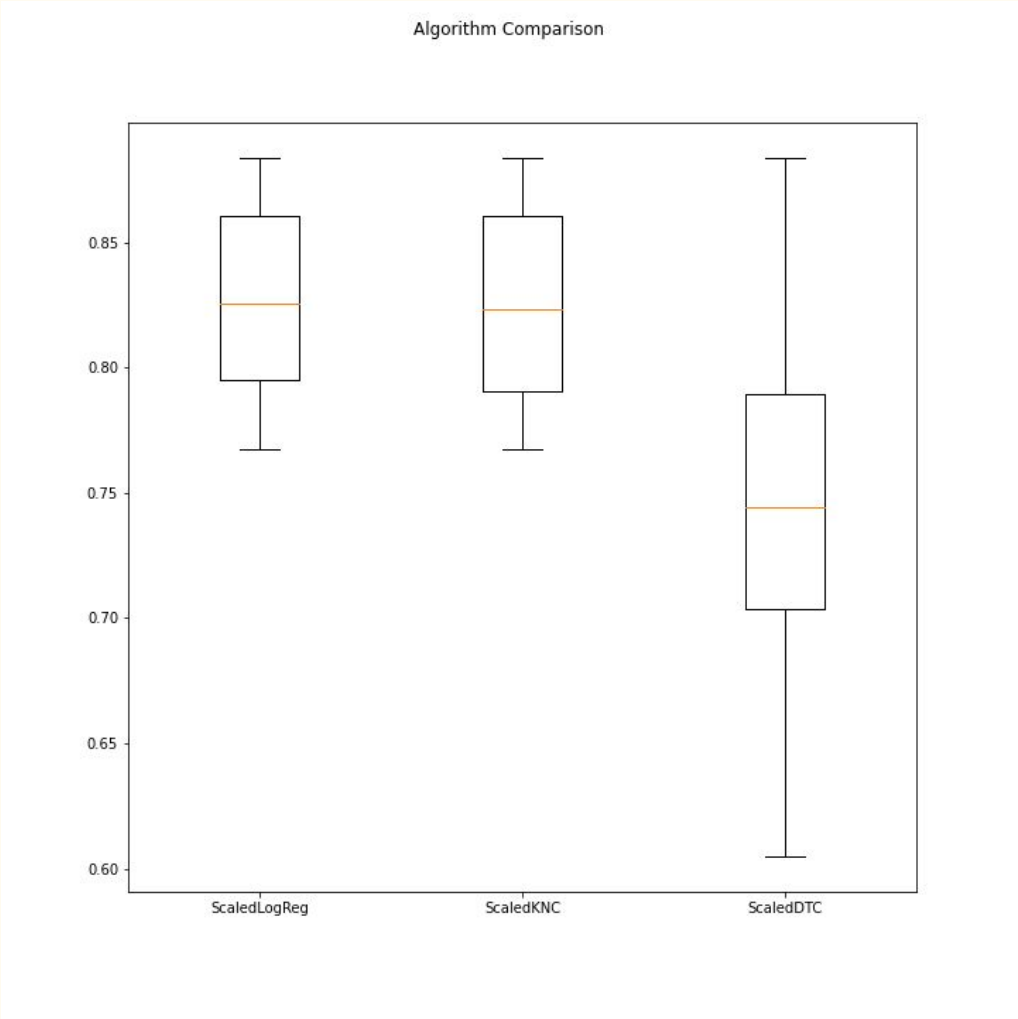
II. ML Model

a. Train model

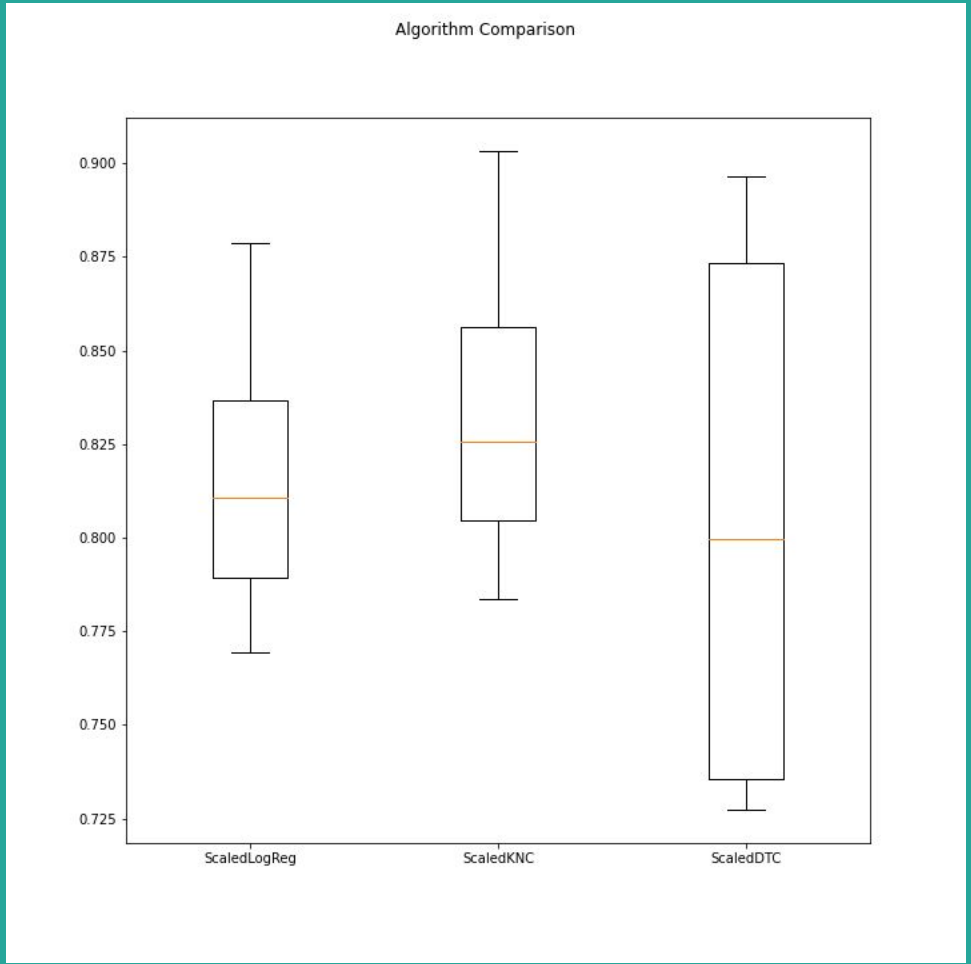
- Do số lượng dữ liệu ít, dùng kfold để train model
- Sử dụng 3 model Logistics Regression, KNClassifier và Decision tree
- Dùng điểm accuracy để đánh giá



Thử train lại model với dữ liệu sau khi scale



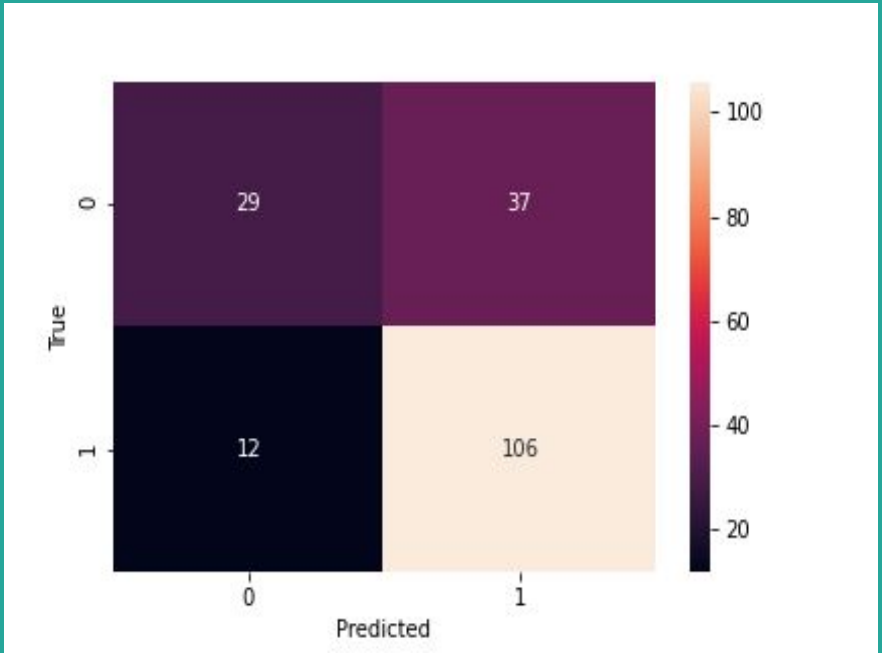
Logistics Regression và KNClassifier cho điểm khá tương đồng, kiểm tra điểm precision



KNC cho điểm precision cao hơn, chọn model này để tối ưu:

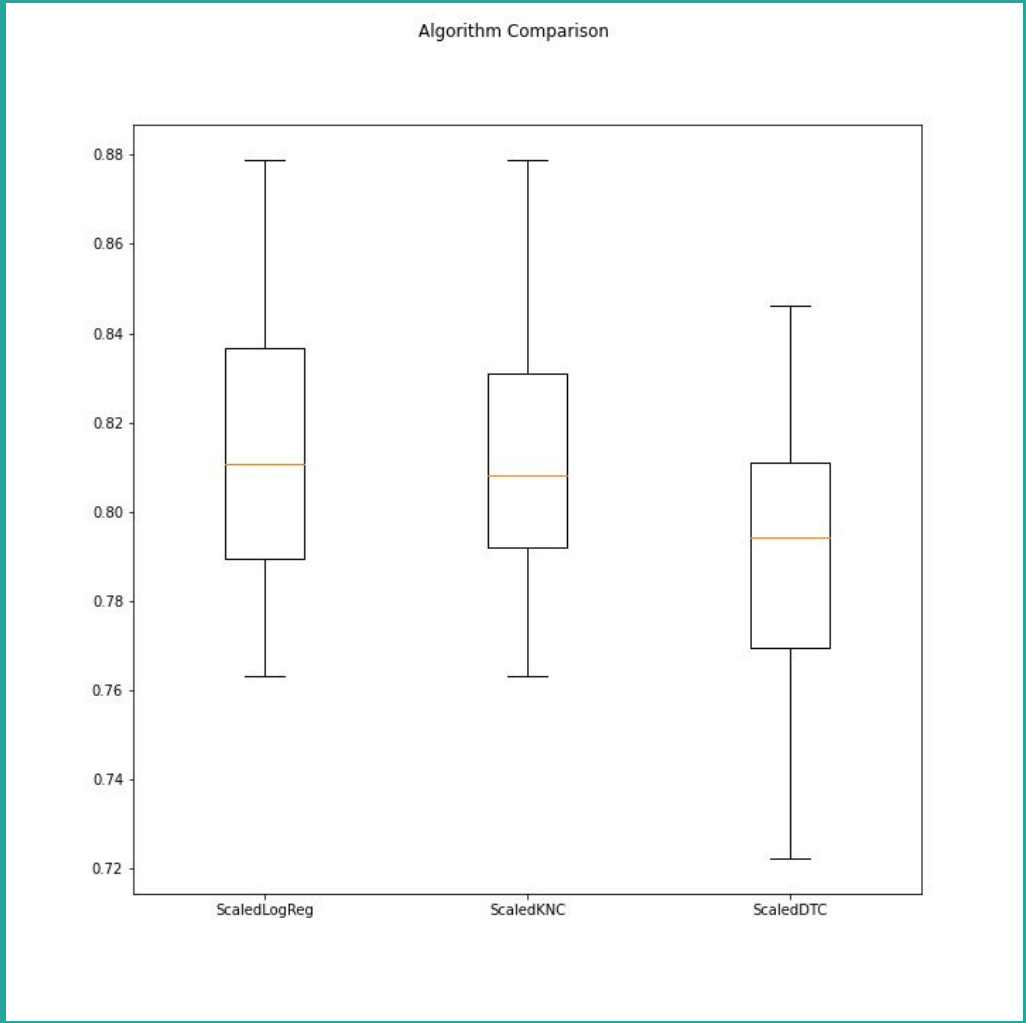
- Tiến hành tune 2 tham số:
  + Metric: euclidean và manhattan
  + N: từ 1 đến 15, step 2

Metric: euclidean và N = 3 cho kết quả tốt nhất, thử dự đoán tập test



	precision	recall	f1-score	support
0	0.71	0.44	0.54	66
1	0.74	0.90	0.81	118
accuracy			0.73	184
macro avg	0.72	0.67	0.68	184
weighted avg	0.73	0.73	0.72	184

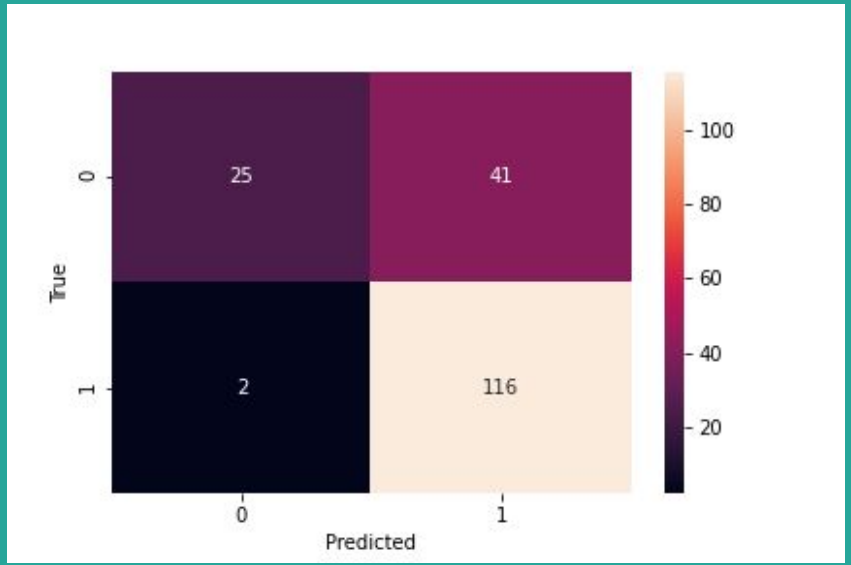
Nhận thấy model chưa tốt lắm, thử chạy lại với bộ dữ liệu đầy đủ, không lọc feature
Dữ liệu đã scale, dùng điểm precision để đánh giá



2 model cho điểm ngang nhau, nhưng Logistics Regression có điểm accuracy cao hơn, dùng model này để tối ưu

- Tiến hành tune 3 tham số:
  + C: 0.001 đến 1000, step nhân 10
  + Max iter: từ 100 đến 500, step bằng 100
  + Solver: newton-cg, lbfgs, liblinear

Kết quả có tốt hơn so với việc lọc feature, thử tune threshold để tối đa điểm precision

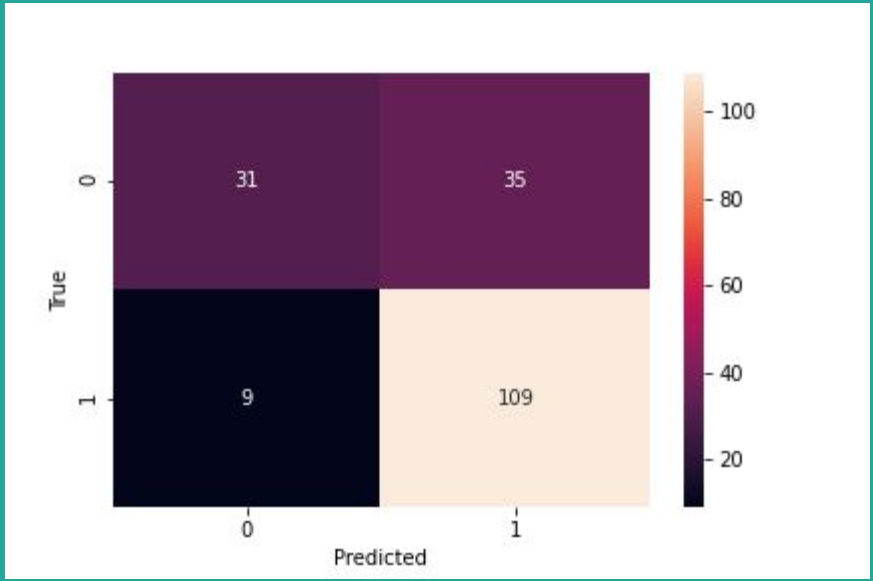


	precision	recall	f1-score	support
0	0.93	0.38	0.54	66
1	0.74	0.98	0.84	118
accuracy			0.77	184
macro avg	0.83	0.68	0.69	184
weighted avg	0.81	0.77	0.73	184

2 model cho điểm ngang nhau, nhưng Logistics Regression có điểm accuracy cao hơn, dùng model này để tối ưu

- Tiến hành tune 3 tham số:
  + C: 0.001 đến 1000, step nhân 10
  + Max iter: từ 100 đến 500, step bằng 100
  + Solver: newton-cg, lbfgs, liblinear

Sau khi chạy thử, thấy threshold = 0.65 cho kết quả tốt nhất do. Chọn model Logistics Regression (C= 10, max iter = 100, solver = lbfgs), threshold = 0.65



	precision	recall	f1-score	support
0	0.78	0.47	0.58	66
1	0.76	0.92	0.83	118
accuracy			0.76	184
macro avg	0.77	0.70	0.71	184
weighted avg	0.76	0.76	0.74	184