ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN:

DỰ ĐOÁN LOẠI NHÀ Ở MELBOURN

GVHD: Nguyễn Thị Anh Thư

Nhóm sinh viên thực hiện:

1. Nguyễn Vũ Anh Khoa	16521511
2. Nguyễn Thanh Bảo	16520086
3. Trương Ngọc Diễm Quyên	16521781
4. Lê Đức Lâm	16520637
5. Ngô Quang Vỷ	16521851
6. Nguyễn Đức Hưng	16520478
7. Lưu Triệu Phương	15520662

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN:

DỰ ĐOÁN LOẠI NHÀ Ở MELBOURN

GVHD: Nguyễn Thị Anh Thư

Nhóm sinh viên thực hiện:

1. Nguyễn Vũ Anh Khoa	16521511
2. Nguyễn Thanh Bảo	16520086
3. Trương Ngọc Diễm Quyên	16521781
4. Lê Đức Lâm	16520637
5. Ngô Quang Vỷ	16521851
6. Nguyễn Đức Hưng	16520478
7. Lưu Triệu Phương	15520662

MỤC LỤC

I. Tìm hiểu dữ liệu

- **Giới thiệu:**
    - Tên data: Melbourne Housing Market
    - Nguồn dữ liệu: Kaggle
    - Link download: https://www.kaggle.com/anthonypino/melbourne-housing-market
- **Tập dữ liệu được thu thập dùng để:**
    - Tập dữ liệu được thu thập từ trên mạng dùng để phân tích thị trường nhà ở Melbourne và các tác vụ liên quan:
        - Xây dựng hệ thống khuyến nghị (Những ngôi nhà nào giá trị? nên mua nhà 2 phòng ngủ ở đâu?...)
        - Dự đoán giá nhà, dự đoán nguyên nhân giá nhà giảm,…
- Tập dữ liệu có **34857** mẫu
- Mỗi mẫu có **20** thuộc tính:

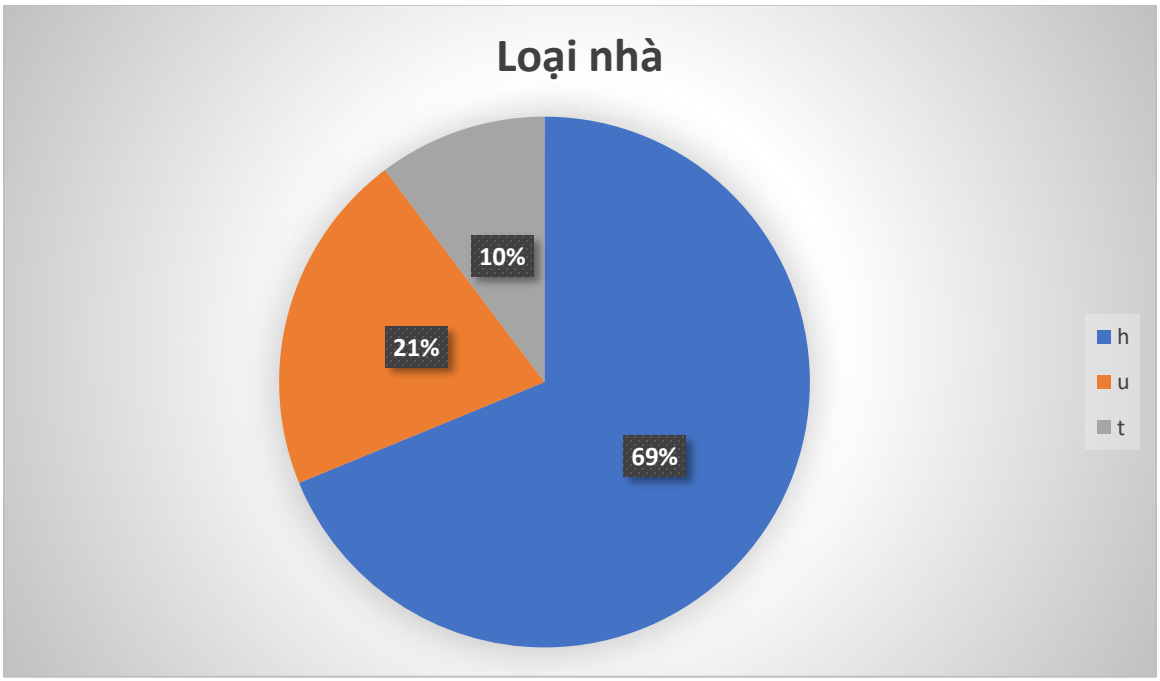
Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Trung bình/số giá trị phân biệt	Phương sai/số giá trị duy nhất	Số mẫu bị thiếu
Suburb	Tên vùng ngoại ô	nominal	351	20	0 (0%)
Address	Địa chỉ	Text	34009	33201	0 (0%)
Rooms	Số phòng	numeric	3.03	0.9409	0 (0%)
Price	Giá	numeric	1050173.34	4.1148E+11	7610 (21.83%)
Method	Phương thức giao dịch	nominal	9	0	0 (0%)

SellerG	Đại lý bất động sản	nominal	388	107	0 (0%)
Date	Ngày bán	nominal	78	0	0 (0%)
Distance	Khoảng cách đến trung tâm tính bằng km	numeric	11.18	46.1041	1 (0%)
Postcode	Mã bưu điện	numeric	3116.06	11885.3604	1 (0%)
Bedroom2	Số phòng ngủ	numeric	3.08	0.9604	8217 (23.57%)
Bathroom	Số phòng tắm	numeric	1.62	0.5184	8226 (23.6%)
Car	Số điểm đậu xe	numeric	1.73	1.0201	8728 (25.04%)
Landsize	Diện tích đất	numeric	593.6	11552113.35	11810 (33.88%)
BuildingArea	Diện tích tòa nhà	numeric	160.26	161017.6129	21115 (60.58%)
YearBuilt	Năm xây dựng	numeric	1965.29	1393.5289	19306 (55.39%)
CouncilArea	Hội đồng quản trị khu vực	nominal	33	0	3 (0.01%)
Lattitude	Vĩ độ	numeric	-37.81	0.0081	7976 (22.88%)
Longtitude	Kinh độ	numeric	145	0.0144	7976 (22.88%)
Regionname	Tên khu vực (đông, tây, nam, bắc,,.)	nominal	8	0	3 (0.01%)
Propertycount	Số lượng bất động sản trong vùng ngoại ô đó	numeric	7572.89	19607981.05	3 (0.01%)

- Nhãn dữ liệu (lables):

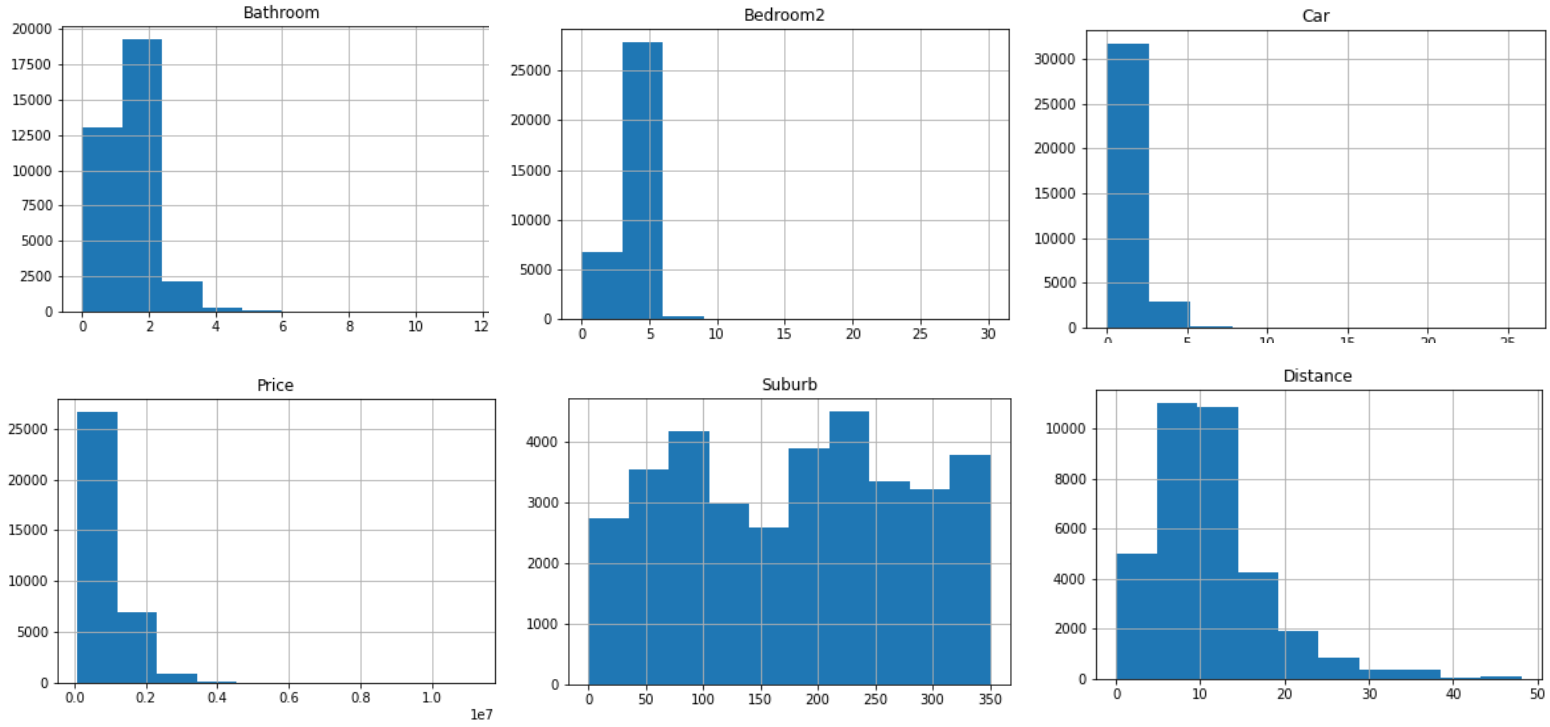
Type	Loại nhà: h - house u - unit t - townhouse	nominal	3	0	0 (0%)

- Kết luận:
    - t/u/h: 3580:7297:23980



    - Dữ liệu bị thiếu nhiều, có nhiều thuộc tính thiếu trên 50% mẫu (YearBuilt, BuildingArea).
    - Miền giá trị giữa các thuộc tính trên lệch cao.
    - Dữ liệu dùng cho nhiều mục đích khác nhau.
- Đây là một tập dữ liệu khó và đầy thử thách để phân tích và áp dụng vào việc phân loại nhà.
- Phân loại nhà để tìm khu vực nào tập trung nhiều căn hộ cao cấp, khu vực nào giá nhà cao.
- Xu hướng nhà ở, vùng nào là tốt nhất để mua từng loại nhà, những nhà nào nằm trong vùng đắt đỏ nhưng có giá cả tốt, nhà nằm ở hướng nào có giá trị cao
- Ứng dụng để tìm khu vực nhà đầu tư để kinh doanh BĐS.

Phân phối (distribution) một số thuộc tính của dữ liệu:



II.Các bước tiền xử lý dữ liệu:

- **Missing value**
  - ✓ Numeric: đối với dữ liệu số sử dụng điền thiếu bằng trung bình của thuộc tính (mean).
  - ✓ Nominal: đối với dữ Nominal/Category sử dụng điền thiếu bằng giá trị xuất hiện nhiều nhất (most frequent) trong thuộc tính đó.
- **Transform**
  - ✓ Nhiệm vụ chính của Transfrom là chuyển tất cả dữ liệu dạng Text/ Nominal/Category về dạng số numeric.
  - ✓ Address: là dữ liệu dạng text nên đầu tiên xây dựng mô hình TF-IDF để chuyển word thành vector, sau đó sử dụng Kmeans (k=8) để phân cụm các vector .

- ✓ Các dữ liệu Nominal/Category chuyển theo index của của tập giá trị duy nhất của thuộc tính đó.

- **Normalization**

  - ✓ Chuẩn hóa các thuộc tính theo min-max về đoạn [-1,1], có sử dụng epsilon = 10^{-9}.

- **Chia dữ liệu**

  - ✓ Bộ dữ liệu được chia theo phương pháp Holdout thành 2 phần theo tỉ lệ 7/3. Trong đó 70% là dữ liệu dùng để train, còn lại 30% dữ liệu dùng để test.

- **Outliers**

  - ✓ Do phân phối (distribution) của các thuộc tính không phải dạng Gussian nên không áp dụng được Standard Deviation để loại nhiễu. Áp dụng phương pháp LocalOutlierFactor để loại bỏ bớt dữ liệu nằm xa luồng dữ liệu chính bằng cách tính khoảng cách giữa các điểm.

III. Chọn Model và cài đặt:

III.1. SVM: Kẻ đường thẳng để phân chia dữ liệu thành 2 phần sao cho khoảng các từ điểm gần nhất gần nhất của dữ liệu đến đường thẳng là lớn nhất.

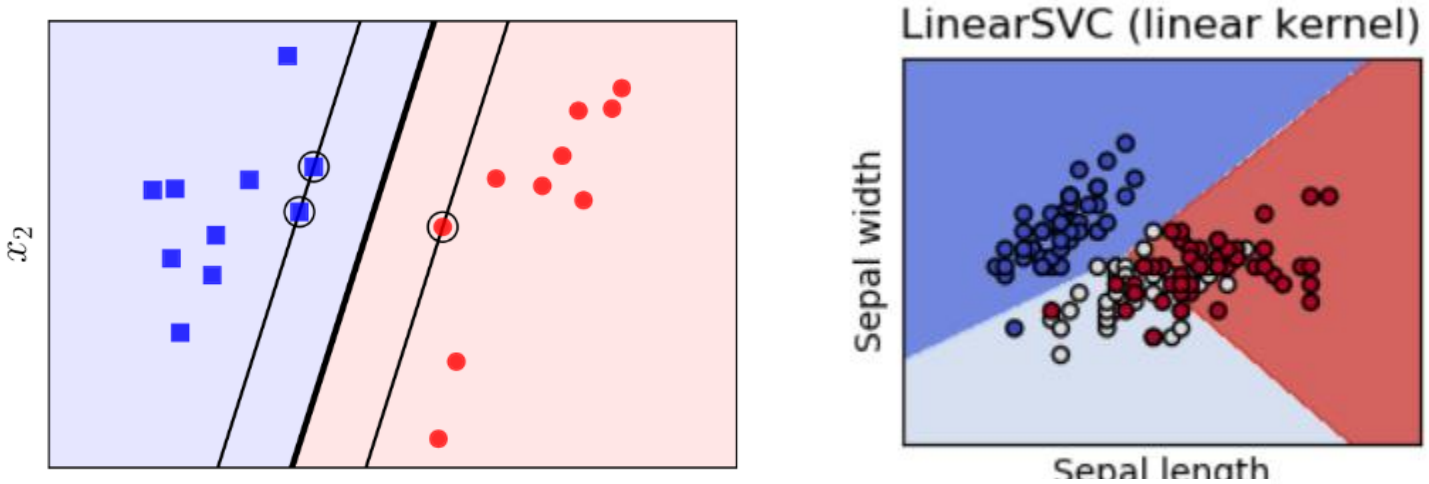


Figure 1 thể hiện thuật toán SVM: bên trái: SVM cho 2 lớp [1], bên phải: SVM cho 3 lớp sử dụng linear kernel [3].

Một số kernel thường dùng:

Kiểu hàm	Công thức
Linear Kernel	K(x,y) = x.y
Polynominal kernel	$K(x,y) = (x.y + 1)^d$
Radial basis function (Gaussian) kernel	$K(x,y) = e^{\frac{-\|x-y\|^2}{2\sigma^2}}$
Hyperbolic tangent kernel	K(x,y) = tanh(a.x.y - b)

Ưu điểm:

- Được sử dụng rộng rãi vì cho ra kết quả tốt đối với những dữ liệu linearly separable.
- Đối với những bài toán nonlinear separable thì có thể sử dụng kernel để đưa về miền dữ liệu mới.

Nhược điểm:

- Đối với SVM không sử dụng kernel hoạt động không hiệu quả đối với dữ liệu nhiễu hoặc dữ liệu gần linearly separable.

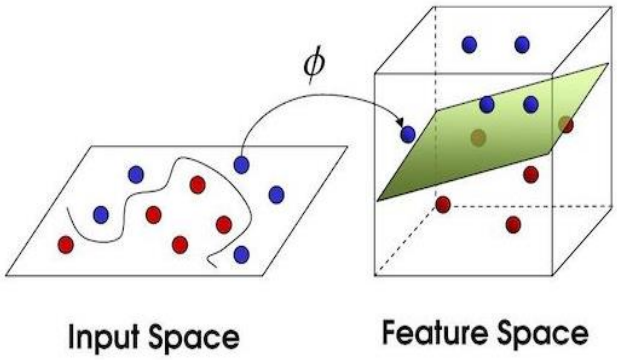
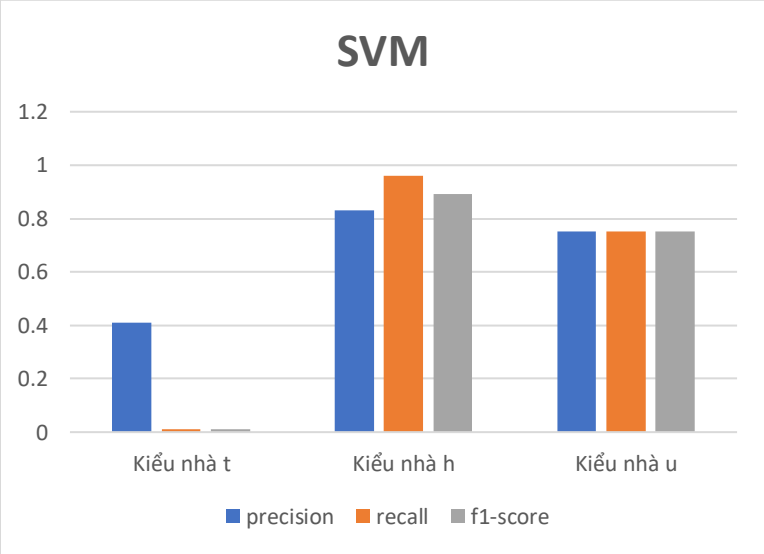


Figure 2 Thể hiện việc áp dụng kernel (nguồn: https://stats.stackexchange.com/questions/18030/how-to-select-kernel-for-svm)

Thực nghiệm

Sử dụng Radial basis function (rbf)

	precision	recall	f1-score	support
t	0.41	0.01	0.01	1133
h	0.83	0.96	0.89	7196
u	0.75	0.75	0.75	2129
accuracy			0.81	10458
macro avg	0.66	0.57	0.55	10458
weighted avg	0.77	0.81	0.76	10458



III.2. Decision tree (cây quyết định)

Ý tưởng: Xây dựng một cây quyết định, tại mỗi nút nội bộ là tên thuộc tính, các nút lá là nhãn của dữ liệu. Tại nút nội bộ ta xét giá trị của thuộc tính đó để phân loại cho bước tiếp theo, nếu đến nút lá nào thì dữ liệu thuộc nhãn đó.

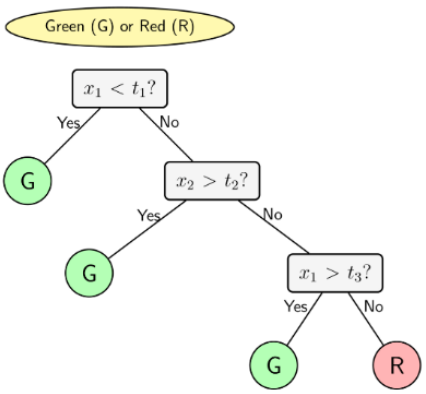


Figure 3 thể hiện thuật toán Decision Tree (nguồn [1])

Có 3 cách xét giá trị thuộc tính:

- Entropy(Information Gain)
- Information Gain Ratio
- Gini Index

Ưu điểm

- Đơn giản để hiểu và giải thích. Cây có thể được hình dung.
- Yêu cầu chuẩn bị dữ liệu ít: dữ liệu đầu vào có thể là là dữ liệu missing, không cần chuẩn hóa hoặc tạo biến giả.

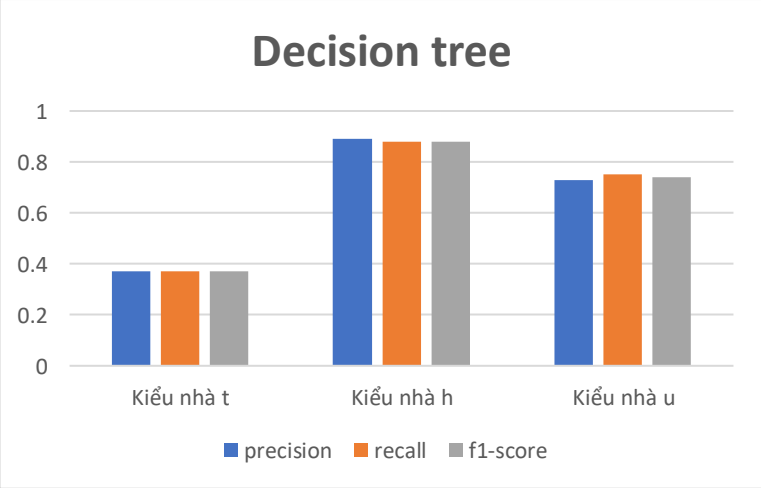
- Là white box model, có thể xác thực mô hình bằng cách sử dụng các kiểm tra thống kê. Ngược lại là black box model (như artificial neural network) khó mà giải thích .
- Có thể làm việc với cả dữ liệu số và dữ liệu phân loại .
- Có khả năng là việc với dữ liệu lớn.

Nhược điểm

- Cây quyết định có thể quá phức tạp, không khái quát hóa tốt dữ liệu hay còn gọi là overfitting. Có thể sử dụng các cơ chế cắt tỉa các nhánh hoạt đặt độ sâu để tránh vấn đề này.
- Cây quyết định có thể không ổn định do mô hình cây quyết định phụ thuộc rất lớn vào dữ liệu đầu vào. Với một sự thay đổi nhỏ trong bộ dữ liệu , cấu trúc mô hình cây quyết định có thể thay đổi hoàn toàn.
- Cây quyết định tạo cây thiên vị nếu một số lớp chiếm ưu thế. Do đó, nên cân bằng tập dữ liệu trước khi khớp với cây quyết định.

Thực nghiệm: Sử dụng Gini index

	precision	recall	f1-score	support
t	0.37	0.37	0.37	1133
h	0.89	0.88	0.88	7196
u	0.73	0.75	0.74	2129
accuracy			0.8	10458
macro avg	0.66	0.67	0.66	10458
weighted avg	0.80	0.80	0.80	10458



III.3. Logistic Regression

Được sử dụng để tính khả năng phân loại [0,1] với đầu vào dữ liệu cụ thể, trong đó thuật toán được biểu diễn dựa trên hàm Logistic Funtion (hàm sigmoid của logarit tự nhiên).

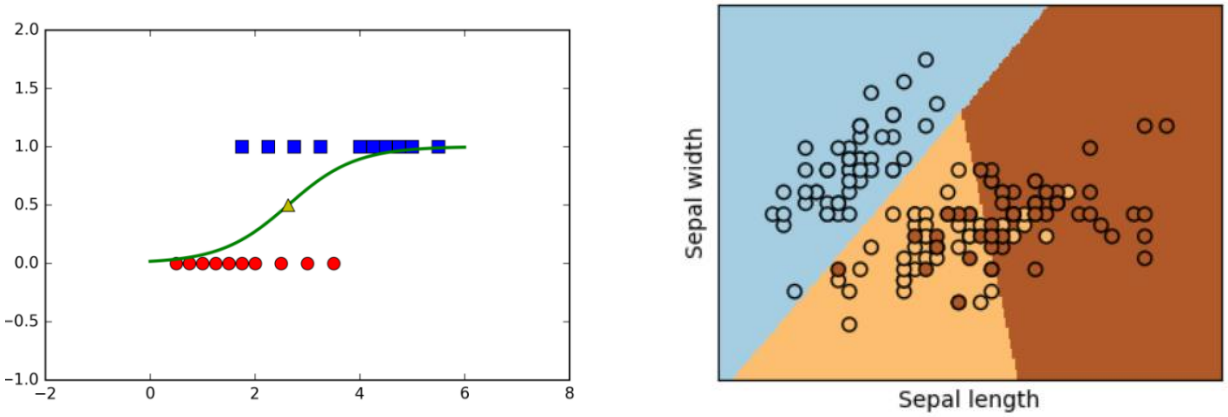


Figure 4 thể hiện thuật toán Logistic Regression: bên trái: Logistic Regression cho 2 lớp [1], bên phải cho 3 lớp [3].

Công thức Logistic Regression có công thức sigmoid: $y = \frac{1}{1+e^{-W^T X}}$

Trong đó:

- y là vector nhãn dữ đoán của tập dữ liệu X.
- X là tập dữ liệu
- W là tham số mà mô hình cần học

Ưu điểm:

- Hoạt động tốt trên dữ liệu linearly separable.
- Logistic Regression ít khi bị overfitting nhưng nó có thể bị overfitting trong các bộ dữ liệu nhiều chiều.
- Logistic Regression dễ thực hiện, diễn giải và rất hiệu quả để đào tạo.

Nhược điểm:

- Giới hạn chính của Logistic Regression là giả định về sự tuyến tính của dữ liệu. Nếu dữ liệu mà ở dạn vòng tròn như figure [5] thì Logistic Regression không làm việc được.
- Nếu số lượng dữ liệu quá ít mẫu thì có thể dẫn đến overfitting.

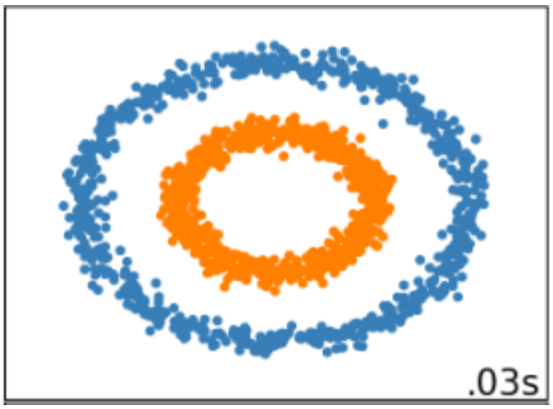
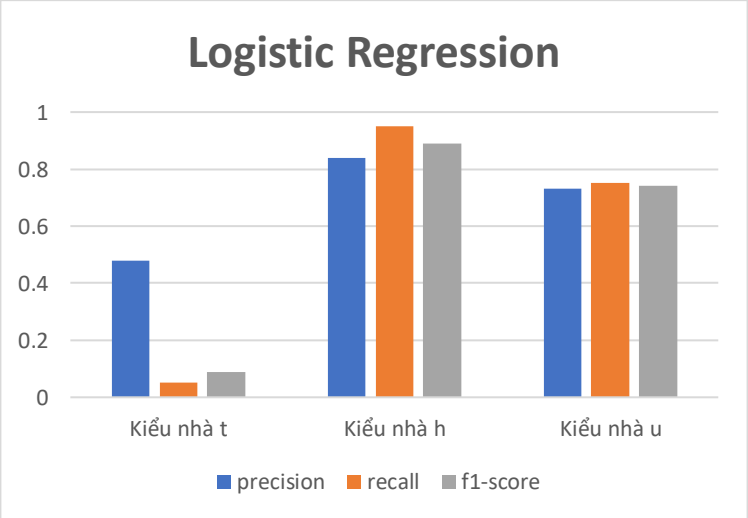


Figure 5 Mô tả tập dữ liệu có 2 nhãn (xanh dương và cam) hình tròn.

Thực nghiệm:

	precision	recall	f1-score	support
t	0.48	0.05	0.09	1133
h	0.84	0.95	0.89	7196
u	0.73	0.75	0.74	2129
accuracy			0.81	10458
macro avg	0.68	0.58	0.57	10458
weighted avg	0.77	0.81	0.77	10458



III.4. Random Forest

Random Forest là một tập hợp của hàng trăm Decision Tree, trong đó mỗi Decision Tree được tạo nên ngẫu nhiên từ việc tái chọn mẫu (chọn random 1 phần của data để xây dựng) và random các biến từ toàn bộ các biến trong trong data. Với một cơ chế như vậy, Random Forest cho ta một kết quả chính xác rất cao nhưng đánh đổi bằng việc ta không thể hiểu cơ chế hoạt động của thuật toán này do cấu trúc quá phức tạp của mô hình này — do vậy thuật toán này là một trong những phương thức Black Box — tức

ta sẽ bỏ tay vào bên trong và rút ra được kết quả chứ không thể giải thích được cơ chế hoạt động của mô hình.

Ưu điểm:

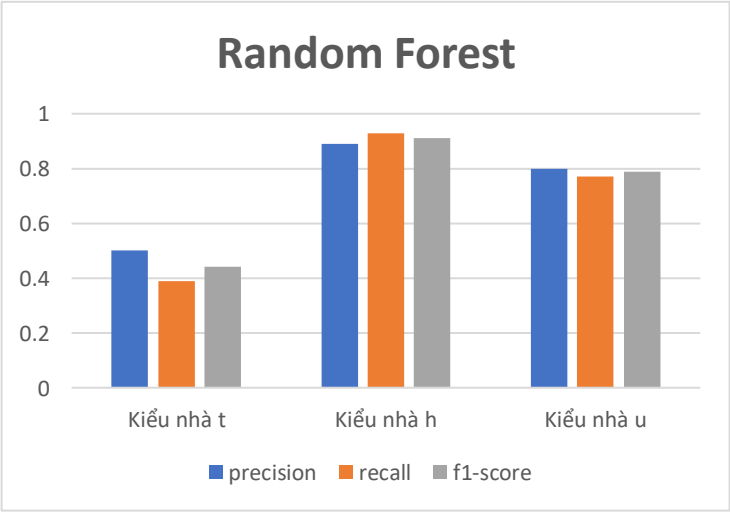
- Random forests được coi là một phương pháp chính xác và mạnh mẽ vì số cây quyết định tham gia vào quá trình này. Nó không bị vấn đề overfitting. Lý do chính là nó mất trung bình của tất cả các dự đoán, trong đó hủy bỏ những thành kiến.
- Thuật toán có thể được sử dụng trong cả hai vấn đề phân loại và hồi quy.
- Random forests cũng có thể xử lý các giá trị còn thiếu. Có hai cách để xử lý các giá trị này: sử dụng các giá trị trung bình để thay thế các biến liên tục và tính toán mức trung bình gần kề của các giá trị bị thiếu

Nhược điểm:

- Random forests chậm tạo dự đoán bởi vì nó có nhiều cây quyết định. Bất cứ khi nào nó đưa ra dự đoán, tất cả các cây trong rừng phải đưa ra dự đoán cho cùng một đầu vào cho trước và sau đó thực hiện bỏ phiếu trên đó. Toàn bộ quá trình này tốn thời gian.

Thực nghiệm

	precision	recall	f1-score	support
t	0.50	0.39	0.44	1133
h	0.89	0.93	0.91	7196
u	0.80	0.77	0.79	2129
accuracy			0.84	10458
macro avg	0.73	0.70	0.71	10458
weighted avg	0.83	0.84	0.83	10458



III.5. Naive bayes

Ý tưởng:

- Gọi D là tập dữ liệu huấn luyện, trong đó mỗi phần tử dữ liệu X được biểu diễn bằng một vector chứa n giá trị thuộc tính $A_1, A_2,...,A_n = \{x_1,x_2,...,x_n\}$
- Giả sử có m lớp $C_1, C_2,..,C_m$. Cho một phần tử dữ liệu X, bộ phân lớp sẽ gán nhãn cho X là lớp có xác suất hậu nghiệm lớn nhất. Cụ thể, bộ phân lớp Bayes sẽ dự đoán X thuộc vào lớp C_i nếu và chỉ nếu:
  $P(C_i|X) > P(C_j|X)$ (1<= i, j <=m, i != j)
  Giá trị này sẽ tính dựa trên định lý Bayes.
- Để tìm xác suất lớn nhất, ta nhận thấy các giá trị P(X) là giống nhau với mọi lớp nên không cần tính. Do đó ta chỉ cần tìm giá trị lớn nhất của $P(X|C_i) * P(C_i)$. Chú ý rằng $P(C_i)$ được ước lượng bằng $|D_i|/|D|$, trong đó D_i là tập các phần tử dữ liệu thuộc lớp C_i. Nếu xác suất tiền nghiệm $P(C_i)$ cũng không xác định được thì ta coi chúng bằng nhau $P(C_1) = P(C_2) = ... = P(C_m)$, khi đó ta chỉ cần tìm giá trị $P(X|C_i)$ lớn nhất.
- Khi số lượng các thuộc tính mô tả dữ liệu là lớn thì chi phí tính toàn $P(X|C_i)$ là rất lớn, dó đó có thể giảm độ phức tạp của thuật toán Naive Bayes giả thiết các thuộc tính độc lập nhau. Khi đó ta có thể tính:
  $P(X|C_i) = P(x_1|C_i)...P(x_n|C_i)$
- Naive Bayes có 3 mô hình chủ yếu:
  - Gaussian Naive Bayes: cho giá trị liên tục
  - Multinomial Naive Bayes: thường được sử dụng cho phân loại văn bảng với mô hình Bag of Word.
  - Bernoulli Naive Bayes: Sử dụng cho giá trị binary 0 ,1

Ưu điểm

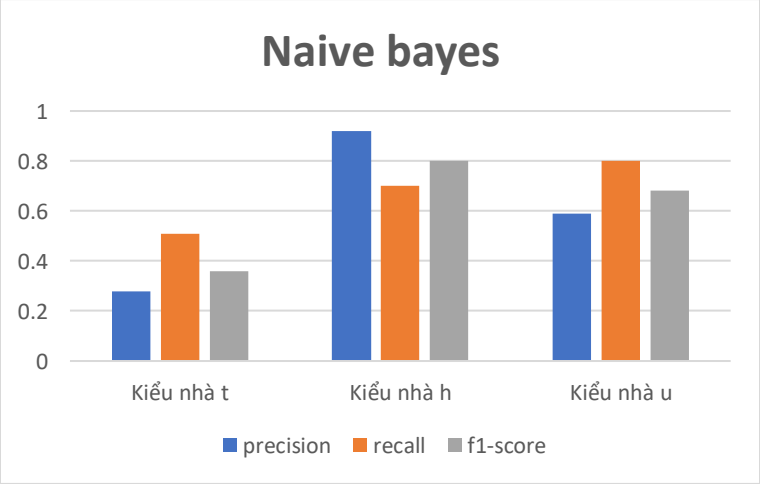
- Giả định độc lập: hoạt động tốt cho nhiều bài toán/miền sữ liệu và ứng dụng.
  Đơn giản nhưng đủ tốt để giải quyết nhiều bài toán như phân lớp văn bản, lọc spam,..
- Cho phép kết hợp tri thức tiền nghiệm (prior knowledge) và dữ liệu quan sát được (obserwed data).
  Tốt khi có sự chệnh lệch số lượng giữa các lớp phân loại.
- Huấn luyện mô hình (ước lượng tham số) dễ và nhanh.

Nhược điểm

- Giả định độc lập (ưu điểm cũng chính là nhược điểm) hầu hết các trường hợp thực tế trong đó có các thuộc tính trong các đối tượng thường phụ thuộc lẫn nhau.
- Mô hình không được huấn luyện bằng phượng pháp tối ưu mạnh và chặt chẽ.
  Tham số mủa mô hình là các ước lượng xác suất điều kiện đơn lẻ.
  Không tính đến sự tương tác giữa các ước lượng này.

Thực nghiệm: sử dụng Gaussian Naive Bayes

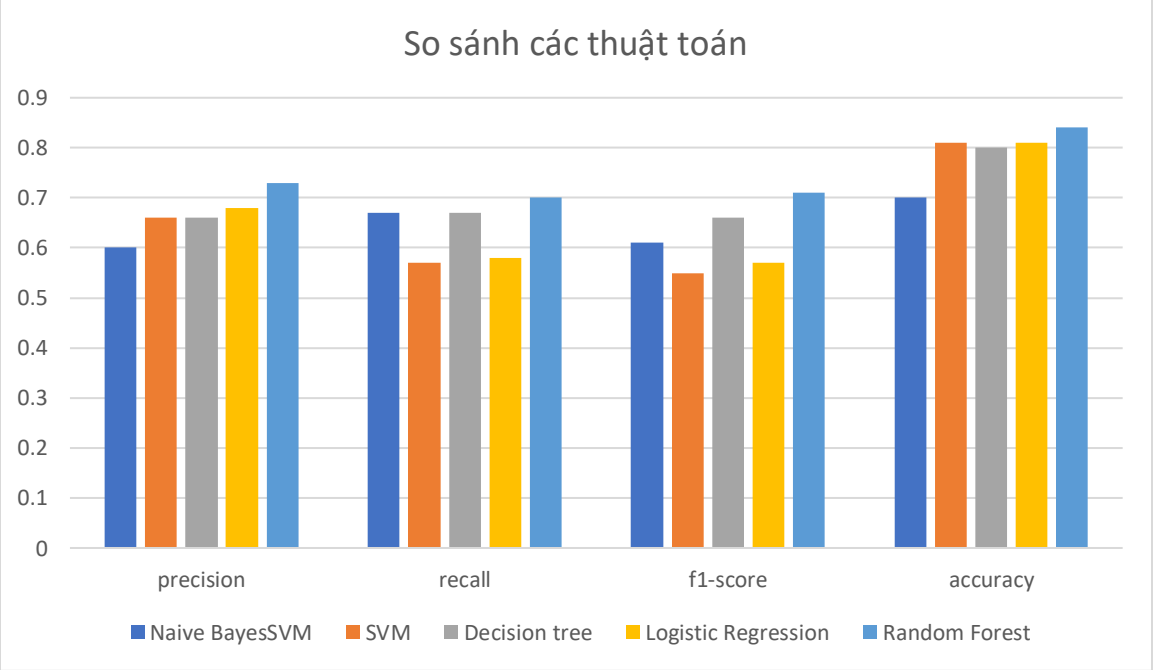
	precision	recall	f1-score	Support
t	0.28	0.51	0.36	1133
h	0.92	0.70	0.80	7196
u	0.59	0.80	0.68	2129
accuracy			0.70	10458
macro avg	0.60	0.67	0.61	10458
weighted avg	0.78	0.70	0.73	10458



IV.Đánh giá

Thuật toán	precision	recall	f1-score	accuracy
Naive Bayes	0.60	0.67	0.61	0.70

SVM	0.66	0.57	0.55	0.81
Decision tree	0.66	0.67	0.66	0.8
Logistic Regression	0.68	0.58	0.57	0.81
Random Forest	**0.73**	**0.70**	**0.71**	**0.84**

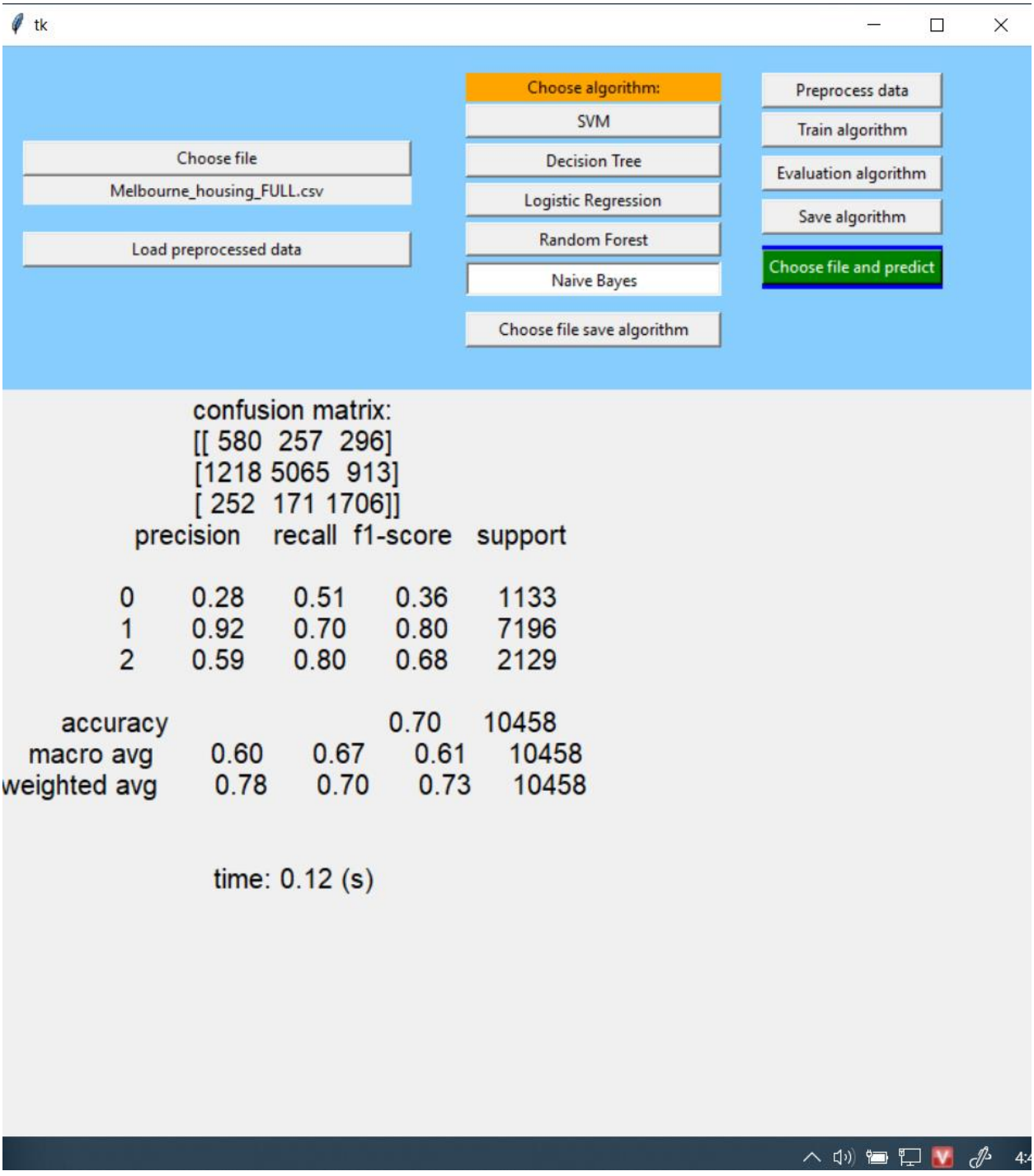


V.Kết luận

- Naive Bayes là thuật toán giả sử các thuộc tính độc lập với nhau nhưng các thuộc tính trong đây có liên qua mật thiết với nhau nên ra kết quả không cao.
- Vì dữ liệu mất cân bằng và có thể không linearly separable nên các thuật toán SVM, Decision tree, Logistic Regression không ra kết quả tốt bằng Random Forest.
- Thuật toán Random Forest đạt được kết quả cao nhất trong các độ đo nên nhóm chúng em chọn Random Forest

VI. Thiết kế giao diện

Sử dụng thư viện tkinter trên python làm giao diện



VII. Tài liệu tham khảo

[1] V. H. Tiệp, "Machine Learning cơ bản." [Online]. Available: https://machinelearningcoban.com/.

[2] Jason Brownlee, " How to Remove Outliers for Machine Learning"[Online]. Available: https://machinelearningmastery.com/how-to-use-statistics-to-identify-outliers-in-data/

[3] “Scikit-learn.” [Online]. Available: scikit-learn.org.