TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN MÔN HỌC

Nhập môn Trí tuệ nhân tạo

Đánh giá mức độ hài lòng của hành khách đi máy bay

Chu Đình Đức
duc.cd194021@sis.hust.edu.vn

Đỗ Minh Hiệp
hiep.dm190048@sis.hust.edu.vn

Đinh Ngọc Huân
huan.dn194065@sis.hust.edu.vn

Ngành: Khoa học máy tính

Giảng viên hướng dẫn: TS. Đỗ Tiến Dũng

HÀ NỘI, 01/2023

LỜI CẢM ƠN

Đồ án môn học Nhập môn Trí tuệ nhân tạo đã được hoàn thiện đầy đủ, chúng em xin dành lời cảm ơn chân thành tới thầy giáo phụ trách môn học TS. Đỗ Tiến Dũng đã có những góp ý quan trọng trong quá trình chúng em thực hiện đồ án, đồng thời đã mang đến những tiết học chất lượng trên giảng đường truyền tải những kiến thức, chia sẻ quý giá đến sinh viên.

Trong quá trình thực hiện đồ án, chúng em đã học hỏi được nhiều kiến thức quý báu, cũng như trau dồi thêm những kỹ năng quan trọng khác. Tuy nhiên đồ án môn học khó tránh khỏi những thiếu sót, chúng em rất mong nhận được nhận xét, chỉ dạy thêm từ thầy để hoàn thiện đồ án, và hoàn thiện bản thân mình trong tương lai.

Em xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Ngày nay, trong một thị trường cạnh tranh ngày một gay gắt thì yếu tố sự hài lòng của khách hàng trở nên cực kỳ quan trọng đối với doanh nghiệp, quyết định sự thành bại của một doanh nghiệp trên thị trường. Việc nâng cao chất lượng dịch vụ, sự hài lòng của khách giúp doanh nghiệp duy trì khách hàng hiện tại, thu hút khách hàng mới, nâng cao lòng trung thành của khách hàng, duy trì và nâng cao khả năng cạnh tranh. Vì vậy trong hoạt động kinh doanh ngày nay việc thỏa mãn khách hàng trở thành hướng đi trọng tâm trong chiến lược kinh doanh của các doanh nghiệp.

Nghiên cứu sự hài lòng của khách hàng giúp doanh nghiệp hiểu được các nhân tố ảnh hưởng đến sự hài lòng và mức độ hài lòng của khách hàng từ đó có thể đánh giá được khả năng cạnh tranh, hiệu quả kinh doanh đồng thời thực hiện các chính sách nhằm khắc phục và nâng cao sự hài lòng của khách hàng đối với sản phẩm, dịch vụ của mình. Đặc biệt, đối với các ngành chuyên về dịch vụ như hàng không, mức độ hài lòng của khách hàng lại càng quan trọng hơn và đóng vai trò quyết định tới sự thành bại của công ty.

1.2 Mô tả bài toán

Từ những phân tích như trên, chúng em đề xuất xây dựng hệ thống *Đánh giá mức độ hài lòng của hành khách đi máy bay* dựa trên lịch sử các chuyến bay trong quá khứ thông qua các mô hình học máy. Lịch sử chuyến bay bao gồm ID khách hàng, giới tính, độ tuổi, loại vé máy bay và một số đánh giá phản hồi từ phía khách hàng về các dịch vụ trong quá trình bay.

Sau khi huấn luyện mô hình học máy, mô hình có thể được dùng để dự đoán mức độ hài lòng của khách hàng đối với những chuyến đi trong tương lai dựa vào những thông tin của chuyến bay tương lai đó.

1.3 Kết quả mong muốn

Các kết quả chúng em mong muốn đạt được trong đồ án môn học:

- Huấn luyện ít nhất hai mô hình nhằm mục đích so sánh kết quả giữa các mô hình với nhau.
- Các mô hình có độ chính xác trên tập dữ liệu test trên 80%.
- Xây dựng một chương trình với giao diện đẹp mắt, đơn giản cho người dùng cuối cùng sử dụng.

CHƯƠNG 2. PHƯƠNG PHÁP ÁP DỤNG

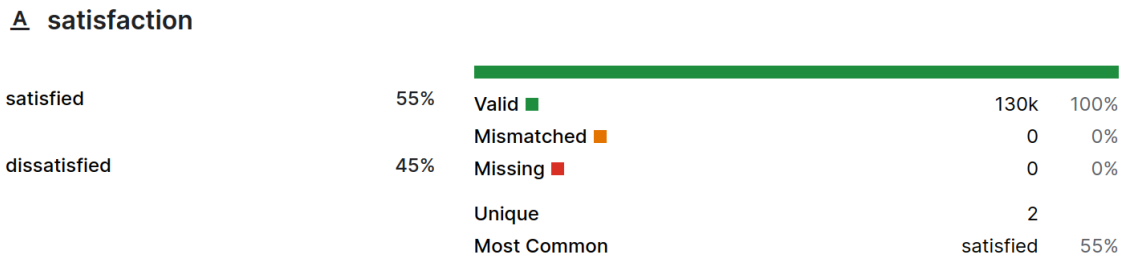
2.1 Tập dữ liệu

Tập dữ liệu sử dụng trong đồ án môn học được đưa ra bởi một tổ chức hàng không. Tên thật của tổ chức không được cung cấp vì nhiều lý do khác nhau. Tập dữ liệu này chứa thông tin chi tiết và phản hồi của những khách hàng đã tham gia các chuyến bay của hãng hàng không này trong quá khứ, trong đó phản hồi của khách hàng trong các bối cảnh khác nhau đã được hợp nhất.

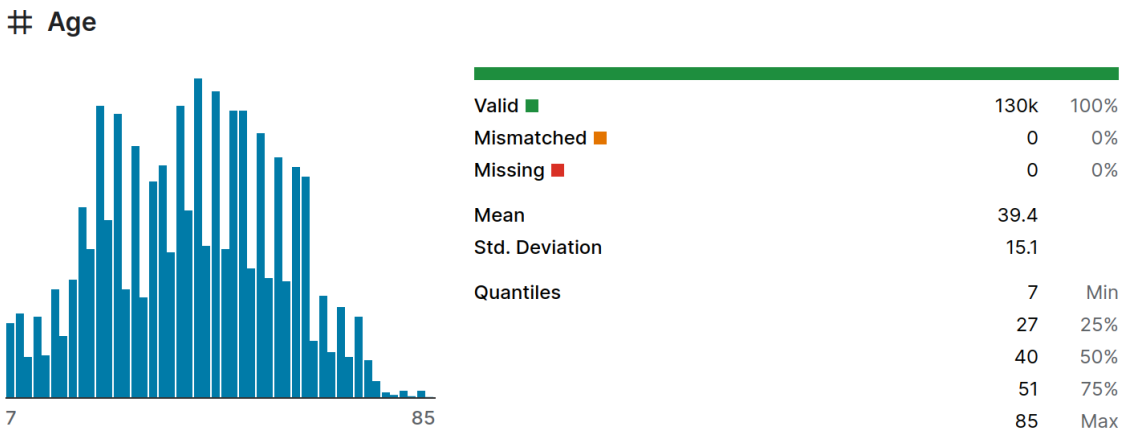
Tập dữ liệu là một file CSV kích thước 12.61 MB gồm 129880 dòng $\times$ 23 cột, tương ứng với 129880 quan sát và mỗi quan sát có 23 thuộc tính. Mô hình cần dự đoán một cách chính xác nhất thuộc tính thứ 23 - khách hàng hài lòng hay không hài lòng dựa vào số liệu của 22 thuộc tính còn lại. Dưới đây là một số thống kê về tập dữ liệu.

satisfaction	0	satisfaction	129880
Gender	0	Gender	129880
Customer Type	0	Customer Type	129880
Age	0	Age	129880
Type of Travel	0	Type of Travel	129880
Class	0	Class	129880
Flight Distance	0	Flight Distance	129880
Seat comfort	0	Seat comfort	129880
Departure/Arrival time convenient	0	Departure/Arrival time convenient	129880
Food and drink	0	Food and drink	129880
Gate location	0	Gate location	129880
Inflight wifi service	0	Inflight wifi service	129880
Inflight entertainment	0	Inflight entertainment	129880
Online support	0	Online support	129880
Ease of Online booking	0	Ease of Online booking	129880
On-board service	0	On-board service	129880
Leg room service	0	Leg room service	129880
Baggage handling	0	Baggage handling	129880
Checkin service	0	Checkin service	129880
Cleanliness	0	Cleanliness	129880
Online boarding	0	Online boarding	129880
Departure Delay in Minutes	0	Departure Delay in Minutes	129880
Arrival Delay in Minutes	393	Arrival Delay in Minutes	129487

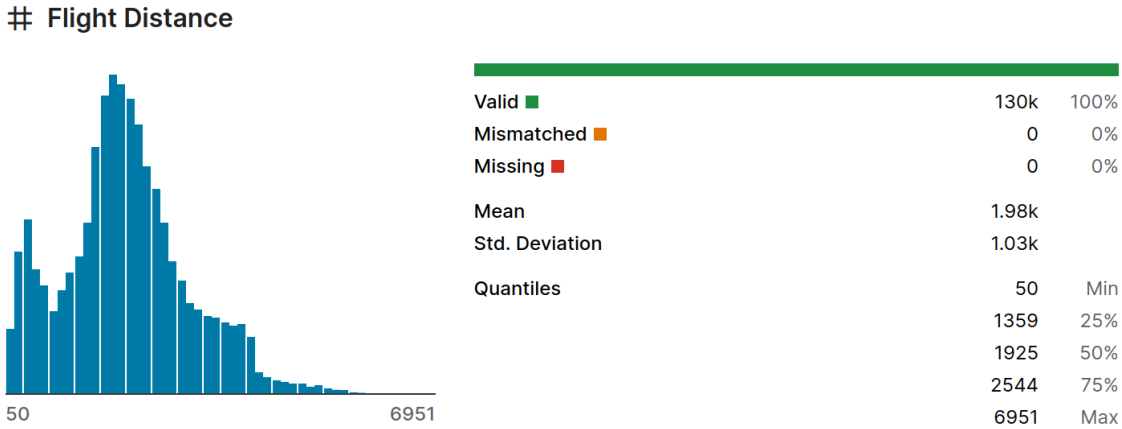
Hình 2.1: Thuộc tính và số lượng giá trị null



Hình 2.2: Số lượng quan sát ở mỗi nhãn



Hình 2.3: Độ tuổi của hành khách



Hình 2.4: Độ dài đường bay

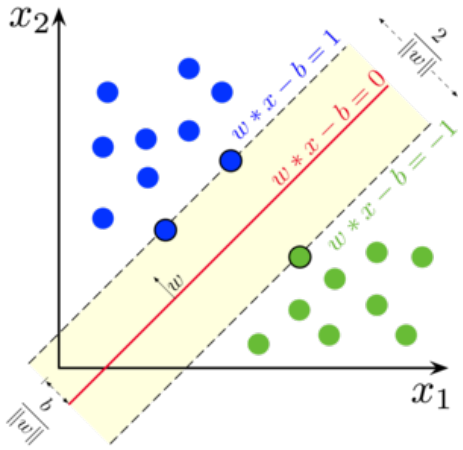
2.2 Mô hình Support Vector Machine

Tổng quan về lớp mô hình SVM và lý do lựa chọn mô hình

Support Vector Machine - SVM là một phương pháp học có giám sát được sử dụng cho các bài toán phân loại, hồi quy và cả các bài toán xác định điểm ngoại lai. Được đề xuất bởi Vapnik và cộng sự những năm 1970, SVM dần trở nên phổ biến vào những năm 1990 và cho tới ngày nay.

Về cơ bản SVM được sử dụng cho bài toán linear classification bằng cách tìm 1 siêu phẳng (được gọi là linear classifier) để chia dữ liệu thành 2 phần. Đối với những tập dữ liệu mà không thể tìm được siêu phẳng nào có thể phân chia tốt, hàm nhân (kernel function) sẽ được sử dụng để chuyển dữ liệu sang không gian chiều khác mà ở đó dữ liệu có thể được phân chia dễ dàng bằng 1 siêu phẳng. *Về lý do lựa chọn mô hình, do hiện tại chúng ta chưa biết liệu rằng có tìm được 1 siêu phẳng có thể phân chia tốt dữ liệu thành 2 phần tương ứng với 2 nhãn (hài lòng hoặc không hài lòng) hay không, do đó trong đồ án môn học lần này mô hình đầu tiên chúng em sẽ triển khai là mô hình LinearSVC - mô hình SVM gốc cho bài toán phân loại*

tương ứng với trường hợp có thể tìm được siêu phẳng như vậy, và mô hình thứ hai là mô hình CatBoost tương ứng với trường hợp chúng ta không thể tìm được siêu phẳng đó. Mô hình CatBoost sẽ được trình bày chi tiết tại phần sau.



Hình 2.5: Support Vector Machine

Những ưu điểm nổi bật của SVM là:

- Hiệu quả trong không gian có số chiều cao
- Vẫn hiệu quả trong trường hợp số chiều lớn hơn số lượng mẫu quan sát
- Chỉ sử dụng 1 tập nhỏ các mẫu quan sát (support vectors) làm bộ phân loại (classifier) nên SVM hiệu quả về bộ nhớ
- Linh hoạt: SVM có nhiều biến thể phù hợp phù hợp với yêu cầu của các bài toán và các tập dữ liệu khác nhau, đặc biệt SVM hoạt động tốt với bài toán phân loại văn bản
- SVM sử dụng lý thuyết đối ngẫu mạnh mẽ trong tối ưu

Bên cạnh đó, SVM cũng có nhược điểm:

- Nếu số chiều lớn hơn số lượng mẫu quan sát sẽ rất khó khăn trong việc lựa chọn biến thể SVM và regulation để tránh bị over-fitting

Lý thuyết chi tiết về SVM

Xét tập training data $D = (x_1, y_1), (x_2, y_2), ..., (x_r, y_r)$ với r quan sát, trong đó:

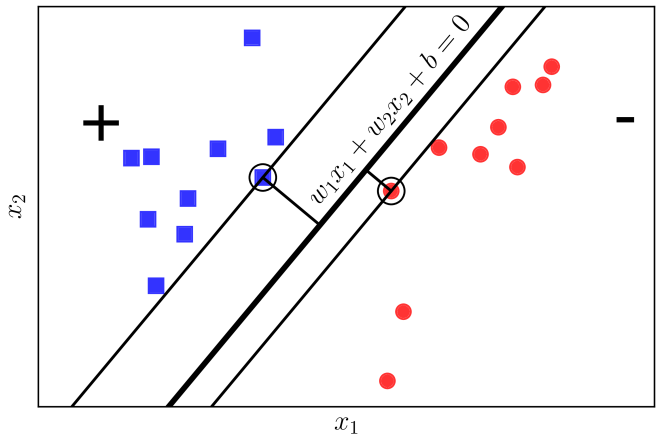
- x_i là vector n chiều
- y_i là nhãn tương ứng của x_i, $y_i \in (-1, 1)$

SVM là một phương pháp giải bài toán phân loại tuyến tính bằng cách tìm một siêu phẳng để chia dữ liệu D thành 2 phần. Giả sử tồn tại 1 siêu phẳng như vậy, siêu phẳng sẽ có dạng:

- $f(x) =< w.x > +b$
- $y_i = 1$ nếu $< w.x_i > +b \geq 0$
- $y_i = -1$ nếu $< w.x_i > +b < 0$

Tuy có nhiều siêu phẳng thỏa mãn nhưng SVM sẽ chọn siêu phẳng có lề lớn nhất (max margin) vì siêu phẳng có lề lớn nhất sẽ có lỗi nhỏ nhất trong số các siêu phẳng có thể có. Ký hiệu $(x^+, 1)$ là điểm trong lớp dương và điểm $(x^-, -1)$ là điểm trong lớp âm sao cho gần với $H_0 :< w.x > +b = 0$ nhất. Hai siêu phẳng lề được định nghĩa như sau:

- H_+ đi qua x^+ và song song với $H_0 :< w.x > +b = 1$
- H_- đi qua x^- và song song với $H_0 :< w.x > +b = -1$
- Và không có điểm dữ liệu nào nằm giữa H_+ và H_-



Hình 2.6: Support Vector Machine

Mức lề (margin) là khoảng cách giữa 2 lề H_+ và H_-:

- $d_+ = d(H_+, H_0) = d(x^+, H_0) = \dfrac{|< w.x^+ > +b|}{\|w\|} = \dfrac{1}{\|w\|}$

- $d_- = d(H_-, H_0) = d(x^-, H_0) = \frac{|< w.x^- > +b|}{\|w\|} = \frac{1}{\|w\|}$
- $margin = d_+ + d_- = \frac{2}{\|w\|}$

Mục tiêu của SVM là học ra được một H_0 với $margin$ lớn nhất, tương đương với bài toán tối ưu có ràng buộc:

$$\max \frac{2}{\|w\|} \Leftrightarrow \min \frac{\|w\|}{2} \text{ conditioned on: } y_i(< w.x_i > +b) \geq 1 \ (*)$$

Bài toán tối ưu trên được giải bằng phương pháp Lagrange như sau:

- Lagrange function: $L(w, b, \alpha) = \frac{1}{2} < w.w > - \sum_{i=1}^{r} \alpha_i[y_i(< w.x_i > +b) - 1]$
- $(*) \Leftrightarrow \underset{w,b}{argmin} \underset{\alpha \geq 0}{max} L(w, b, \alpha) = \underset{w,b}{argmin} \underset{\alpha \geq 0}{max}(\frac{1}{2} < w.w > - \sum_{i=1}^{r} \alpha_i[y_i(< w.x_i > +b) - 1]$
- Primal problem: $\underset{\alpha \geq 0}{max} L(w, b, \alpha) = \underset{\alpha \geq 0}{max}(\frac{1}{2} < w.w > - \sum_{i=1}^{r} \alpha_i[y_i(< w.x_i > +b) - 1])$
- Dual problem: $\underset{w,b}{min} L(w, b, \alpha) = \underset{w,b}{min}(\frac{1}{2} < w.w > - \sum_{i=1}^{r} \alpha_i[y_i(< w.x_i > +b) - 1])$

Nghiệm tối ưu của bài toán thỏa mãn điều kiện KKT:

- $\frac{\partial L}{\partial w} = w - \sum_{i=1}^{r} \alpha_i y_i x_i = 0$
- $\frac{\partial L}{\partial b} = - \sum_{i=1}^{r} \alpha_i y_i = 0$
- $y_i(< w.x_i > +b) - 1 \geq 0 \ \forall x_i (i = 1..r)$
- $\alpha_i \geq 0$
- $\alpha_i(y_i(< w.x_i > +b) - 1) = 0$

Bằng các tính đạo hàm của $L(w, b, \alpha)$ biến (w,b) và cho đạo hàm bằng 0 thu được một hàm đối ngẫu. Bài toán gốc tương đương với bài toán:

$$\text{Maximize } L_D(\alpha) = \sum_{i=1}^{r} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{r} \alpha_i \alpha_j y_i y_j < x_i.x_j >$$

$$\text{such that: } \sum_{i=1}^{r} \alpha_i y_i = 0, \ \alpha_i \geq 0 \ \forall i = 1..r$$

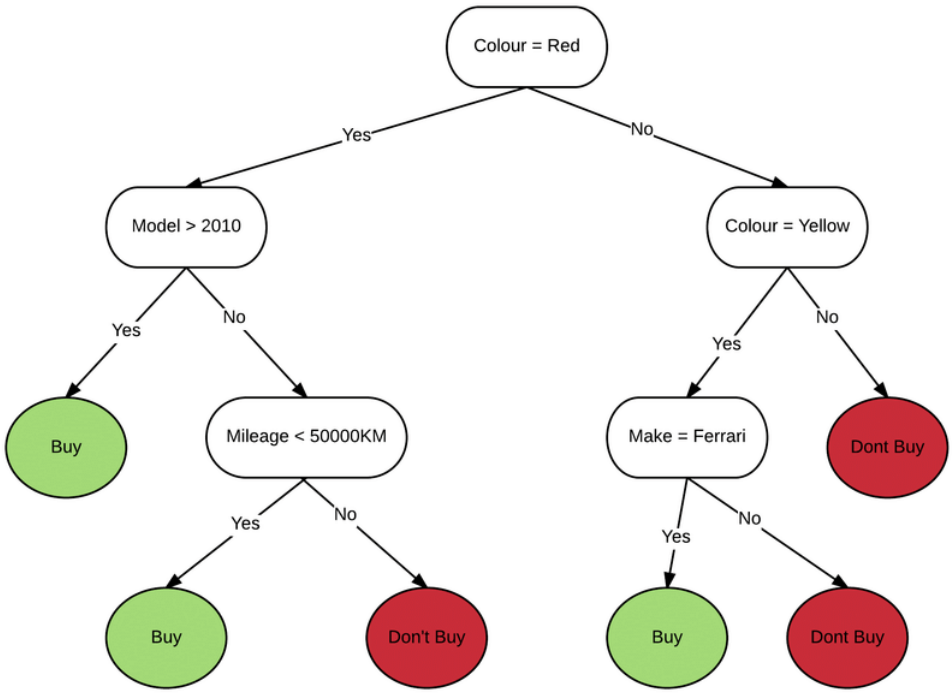
Từ điều kiện ngay phía trên ta nhận thấy nếu $\alpha_i > 0$ thì $x_i > 0$ là một vector hỗ trợ. Gọi SV là tập tất cả các vector hỗ trợ (SV là tập con của training data). w^* và b được tính dựa vào điều kiện KKT như sau:

- $w^* = \sum_{i=1}^{r} \alpha_i y_i x_i = \sum_{x_i \in SV} \alpha_i y_i$
- Với $\alpha_k > 0$ ta có $y_k(< w^*.x_k > +b*) - 1 = 0 \Leftrightarrow b^* = y_k - < w^*.x_k >$

2.3 Mô hình CatBoost

CatBoost là một thuật toán sử dụng gradient booosting trên cây quyết định được phát triển năm 2017 bởi Yandex - 1 công ty chuyên về công cụ tìm kiếm của Nga.

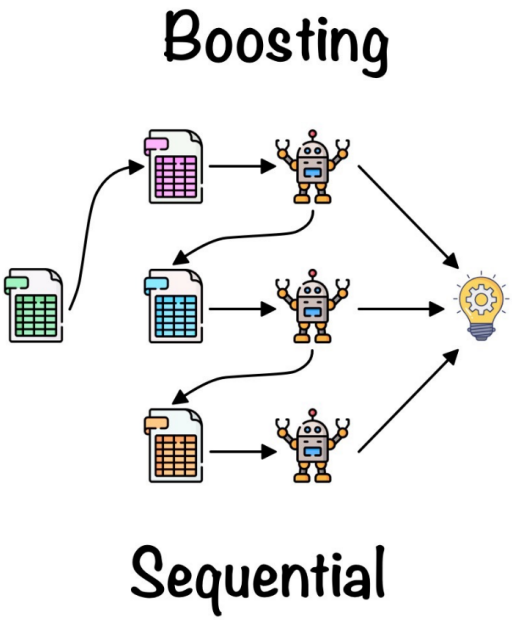
Cây quyết định là một dạng cấu trúc dữ liệu được ứng dụng trong nhiều lĩnh vực. Một cây thông thường sẽ có 3 loại node: node gốc, node quyết định và node lá. Trong học máy, cây quyết định được sử dụng cho cả bài toán phân loại (classification problem) và bài toán hồi quy (regression problem). Khi một cây đã được học, chúng ta có thể dự đoán nhãn của một quan sát mới theo cách đi từ gốc tới lá của cây thông qua các thuộc tính của quan sát.



Hình 2.7: Cây quyết định

CatBoost là một biến thể nổi bật của thuật toán Gradient Boosting. Vì vậy trước tiên hãy cùng tìm hiểu về Gradient Boosting và sau đó là xem xét một số ưu điểm của CatBoost để hiểu lý do tại sao chúng ta lại chọn mô hình CatBoost.

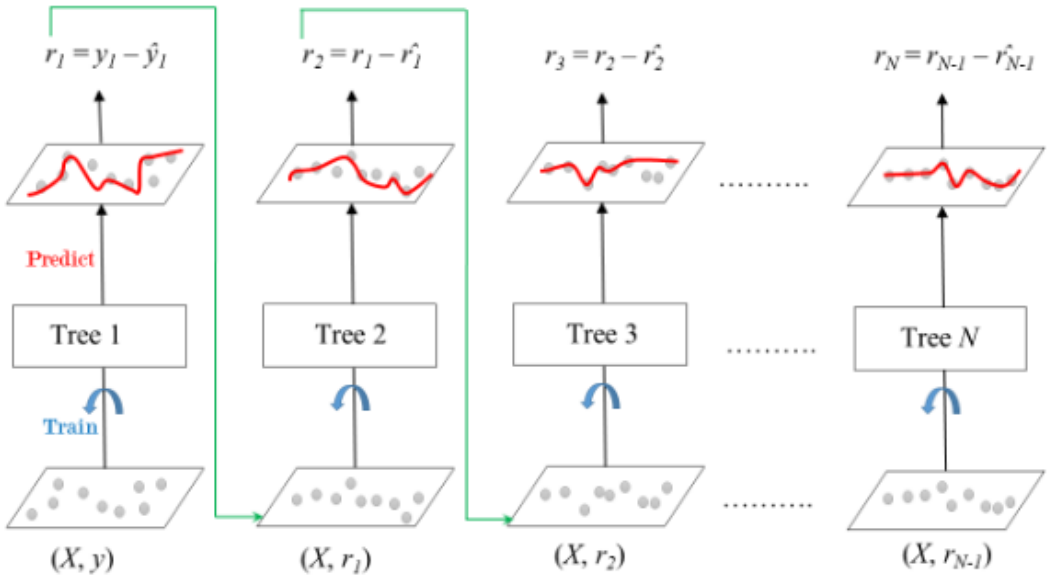
Boosting là lớp phương pháp ensemble learning, lớp phương pháp này tổng hợp nhiều model yếu thành một model mạnh hơn - sức mạnh của sự đoàn kết. Cụ thể trong Boosting, tập dữ liệu được học bởi các cây khác nhau theo cách tuần tự nghĩa là cây sau sẽ được học tiếp từ cây trước đó với mong muốn rằng cây phía sau sẽ hoàn thiện được những yếu điểm của cây phía trước. Cuối cùng chúng ta sẽ tổng hợp kết quả từ tất cả các cây đã học bằng cách tổng hợp có trọng số hoặc tổng hợp trung bình.



Hình 2.8: Lớp phương pháp Boosting

Gradient Boosting là một phương pháp Boosting được hiểu một cách cơ bản theo các bước sau:

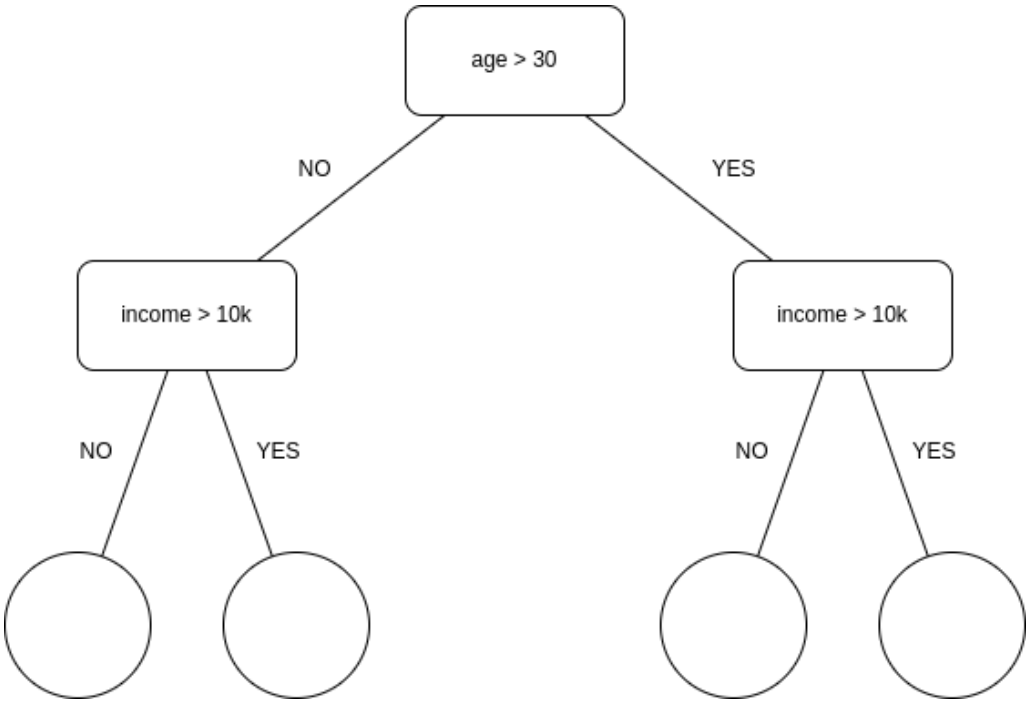
- Bước 1: Huấn luyện cây quyết định bằng tập train
- Bước 2: Dùng cây đã huấn luyện để dự đoán trên tập train
- Bước 3: Tính *error* tại mỗi điểm dữ liệu trong tập train: $r_i = y_i - \hat{y}_i$
- Bước 4: Xem r_i là nhãn của tập train và lặp lại bước 1 cho tới khi đạt đến số lượng cây nhất định hoặc $error$ giảm không đáng kể



Hình 2.9: Phương pháp Gradient Boosting

CatBoost là phiên bản cải tiến của Gradient Boosting. Những cải tiến của CatBoost so với Gradient Boosting và các biến thể khác của Gradient Boosting làm cho CatBoost trở nên vượt trội hơn bao gồm:

- *Cây cân bằng:* Không giống như XGBoost hay LightGBM, CatBoost xây dựng nên các cây cân bằng (cây đối xứng). Mọi node tại cùng level sử dụng chung một cặp thuộc tính - ngưỡng với loss thấp nhất để phân nhánh. Kiến trúc cây cân bằng này hiệu quả khi triển khai với CPU, giảm thời gian dự đoán, áp dụng model nhanh chóng và tránh overfitting do cấu trúc cây cũng được xem như một kiểu regulation.
- *Boosting có thứ tự:* Các thuật toán Boosting cổ điển thường có xu hướng overfitting trên tập dữ liệu nhỏ/nhiễu do các cây được train trên cùng một bộ dữ liệu. CatBoost trước tiên sẽ hoán vị dữ liệu sau đó chia dữ liệu thành 2 phần 1 phần để train và 1 phần để đánh giá để tránh những yếu điểm trên.
- *Hỗ trợ thuộc tính gốc:* CatBoost hỗ trợ hầu hết các kiểu thuộc tính như numeric (số), categorical (chữ), text (văn bản) giúp giảm thời gian và công sức tiền xử lý



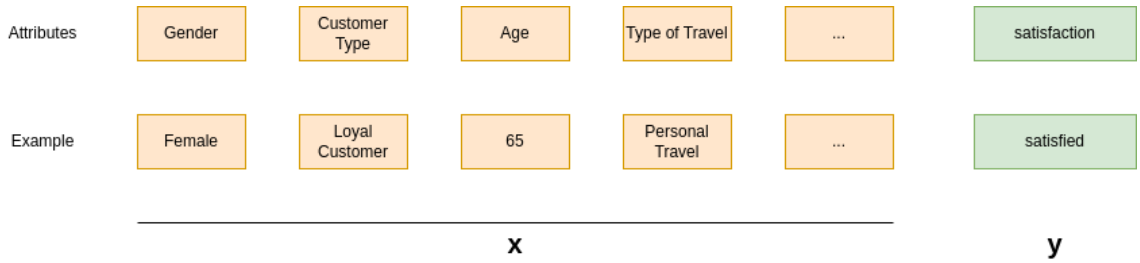
Hình 2.10: Cây CatBoost

CHƯƠNG 3. XÂY DỰNG CHƯƠNG TRÌNH

3.1 Tiền xử lý dữ liệu

Bước đầu tiên và cũng là bước quan trọng nhất khi triển khai các mô hình học máy là tiền xử lý dữ liệu. Đây là gian đoạn rất tốn kém thời gian, thường chiếm khoảng 70 - 90% tổng thời gian của dự án. Tuy nhiên tiền xử lý dữ liệu trước khi huấn luyện model giúp chúng ta có một bộ dữ liệu tốt để học ra một model chất lượng cao.

Tập dữ liệu huấn luyện là tệp CSV (Comma Separated Value) gồm 129880 dòng $\times$ 23 cột. Có thể hiểu hiện tại chúng ta có 129880 cặp (x, y), trong đó nhãn y là vector 1 chiều tương ứng với 1 thuộc tính cần dự đoán trong 23 thuộc tính và x là vector 22 chiều tương ứng với 22 thuộc tính còn lại.



Hình 3.1: Cấu trúc dữ liệu huấn luyện

Bước đầu, chúng em thực hiện loại bỏ (drop) tất cả các hàng có chứa giá trị null. Quan thống kê tại Hình 2.1 chỉ có duy nhất một thuộc tính *Arrival Delay in Minutes* chứa giá trị null, từ đó chúng em đã tiến hành loại bỏ 393 hàng chứa giá trị null tại thuộc tính *Arrival Delay in Minutes*, chiếm tỉ lệ $393/129880 \approx 0.3\%$ trên tổng số lượng mẫu quan sát.

Bước tiếp theo, chúng em thực hiện mã hóa các thuộc tính dạng chữ sang dạng số vì mô hình LinearSVC chỉ làm việc được với dữ liệu dạng số. Ngoài ra chỉ có 5/23 thuộc tính dạng chữ còn lại là 22 thuộc tính dạng số, chúng em tin rằng chuyển 5 thuộc tính chữ về thuộc tính số sẽ giúp cho mô hình học tốt hơn ngay cả khi sử dụng mô hình CatBoost - mô hình này có thể được huấn luyện trực tiếp từ các thuộc tính chữ mà không cần thực hiện bước mã hóa này. Vì vậy, 5 thuộc tính chữ được mã hóa như sau:

- *satisfaction:* satisfied - 1, dissatisfied - 0
- *Gender:* Female - 1, Male - 0
- *Customer Type:* Loyal Customer - 1, disloyal Customer - 0

- *Type of Travel:* Business travel - 2, Personal Travel - 1
- *Class:* Business - 3, Eco Plus - 2, Eco - 1

Bước tiếp, nhận thấy độ lớn về giá trị của các thuộc tính khác nhau có sự chênh lệnh lớn, hãy xem xét bảng sau.

satisfaction	2	[satisfied, dissatisfied]
Gender	2	[Female, Male]
Customer Type	2	[Loyal Customer, disloyal Customer]
Age	75	[65, 47, 15, 60, 70, 30, 66, 10, 56, 22, 58, 3...
Type of Travel	2	[Personal Travel, Business travel]
Class	3	[Eco, Business, Eco Plus]
Flight Distance	5398	[265, 2464, 2138, 623, 354, 1894, 227, 1812, 7...
Seat comfort	6	[0, 1, 4, 5, 2, 3]
Departure/Arrival time convenient	6	[0, 1, 2, 3, 4, 5]
Food and drink	6	[0, 1, 2, 3, 4, 5]
Gate location	6	[2, 3, 4, 1, 5, 0]
Inflight wifi service	6	[2, 0, 3, 4, 5, 1]
Inflight entertainment	6	[4, 2, 0, 3, 5, 1]
Online support	6	[2, 3, 4, 5, 1, 0]
Ease of Online booking	6	[3, 2, 1, 5, 4, 0]
On-board service	6	[3, 4, 1, 2, 5, 0]
Leg room service	6	[0, 4, 3, 2, 5, 1]
Baggage handling	5	[3, 4, 1, 2, 5]
Checkin service	6	[5, 2, 4, 3, 1, 0]
Cleanliness	6	[3, 4, 1, 2, 5, 0]
Online boarding	6	[2, 3, 5, 4, 1, 0]
Departure Delay in Minutes	466	[0, 310, 17, 30, 47, 40, 5, 2, 34, 4, 13, 427,...
Arrival Delay in Minutes	472	[0.0, 305.0, 15.0, 26.0, 48.0, 23.0, 19.0, 2.0...

Hình 3.2: Độ lớn của giá trị các thuộc tính

Nhận thấy các thuộc tính như *Food and drink*, *Gate location* có giá trị nằm trong đoạn $[0, 6]$ nhưng các thuộc tính như *Age*, *Flight Distance* có giá trị lớn hơn nhiều, điều này làm cho mô hình sẽ quan tâm nhiều hơn đến các thuộc tính có giá trị lớn và quan tâm ít hơn đến các thuộc tính có giá trị nhỏ, dẫn đến mô hình học không chuẩn. Do đó chúng em thực hiện công đoạn scale dữ liệu để thu được bộ dữ liệu đồng nhất thông qua phương pháp scale chuẩn - Standard Scale.

$$z = \frac{x - \mu}{\sigma} \text{ với } \mu = \frac{1}{N}\sum_{i=1}^{N} x_i \text{ và } \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

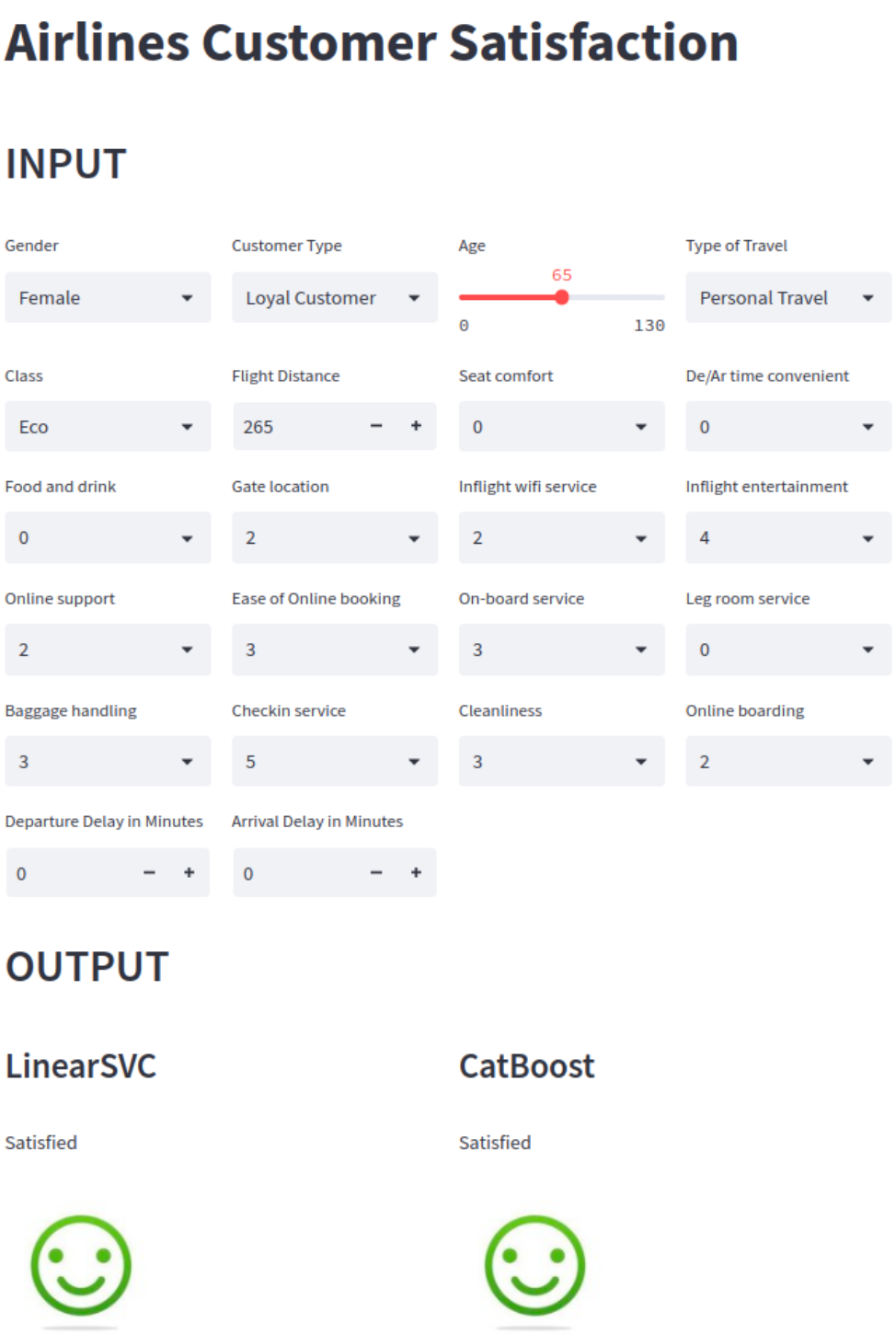
Và bước cuối cùng, chúng em kiểm tra thấy rằng tập dữ liệu này cân bằng đối với nhãn (55% satisfied và 45% dissatisfied) do đó chúng em sẽ không cần phải thực hiện thêm công đoạn để giải quyết vấn đề cân bằng dữ liệu.

Sau khi đã tiền xử lý dữ liệu, dữ liệu được sử dụng để train hai model LinearSVC và CatBoost. Khi quá trình train model hoàn thành, chúng em sẽ lưu model và sử dụng model này dự đoán kết trên tập test để đánh giá chất lượng của model thông qua các độ đo như accuracy, precision, recall, F1 score, ... ngoài ra model còn được sử dụng cho công đoạn demo đồ án môn học bằng framework Streamlit - một

framework được sử dụng để triển khai các model học máy. Trước khi thực hiện huấn luyện mô hình, chúng em đã thử nhiều bộ tham số và chọn ra bộ tham số phù hợp nhất bằng một số kỹ thuật riêng biệt, và sử dụng bộ tham số này để huấn luyện mô hình. Tập dữ liệu train - test được chia theo tỷ lệ 0.8 - 0.2.

3.2 Chương trình

Chương trình được viết theo framework Streamlit. Sau đây là giao diện tổng quan của chương trình.



Hình 3.3: Giao diện của chương trình

Input đầu vào là giá trị của 22 thuộc tính và output đầu ra là kết quả dự đoán của hai mô hình, bên trái LinearSVC và bên phải CatBoost, đầu ra sẽ là *Satisfied* nếu mô hình dự đoán khách hàng sẽ hài lòng và là *Dissatisfied* nếu mô hình dự đoán khách hàng sẽ không hài lòng, kèm theo icon biểu diễn trạng thái tương ứng.

Khi thực hiện dự đoán, người dùng chỉ cần chọn giá trị hoặc nhập giá trị tương ứng của các thuộc tính input, chương trình sẽ tự động hiển thị kết quả ở phần output có thể xem như ngay lập tức mà không có độ trễ. Khi đã dự đoán xong, nếu người dùng thay đổi giá trị tại bất kỳ thuộc tính input nào, mô hình sẽ tự động tính toán lại kết quả dự đoán đối với input hiện tại và hiển thị lại kết quả.

3.3 Khó khăn gặp phải và cách giải quyết

Khó khăn đầu tiên chúng em gặp phải là vấn đề lựa chọn mô hình sao cho phù hợp và hiệu quả. Lựa chọn mô hình phù hợp với tập dữ liệu là vấn đề chung của các dự án học máy và rất khó để đưa ra câu trả lời chính xác cho câu hỏi này. Sau thời gian tìm hiểu và trao đổi giữa các thành viên trong nhóm, chúng em quyết định sẽ thử nghiệm 2 mô hình - LinearSVC và CatBoost để so sánh kết quả cũng như sự hiệu quả giữa chúng. Vì theo cảm quan của chúng em, mô hình LinearSVC sẽ hiệu quả nếu như tập dữ liệu là linear theo thuộc tính và mô hình CatBoost sẽ hiệu quả trong trường hợp còn lại - tập dữ liệu là non-linear. Lý do cụ thể về việc lựa chọn mô mình đã được chúng em đề cập tại mục 2.2.

Ngoài ra, chúng em cũng gặp khó khăn khi viết chương trình triển khai đồ án theo framework Streamlit. Trong quá trình học tập, chúng em đã tập trung và dành nhiều thời gian cho giai đoạn xử lý dữ liệu và huấn luyện mô hình mà chưa có kinh nghiệm trong việc triển khai mô hình, xây dựng giao diện để cho người dùng sử dụng và trước mắt là để demo trước thầy và các bạn trong lớp học. Tuy nhiên, nhờ sự nỗ lực cố gắng của các thành viên trong nhóm, chúng em đã tìm hiểu về Streamlit và sử dụng Streamlit để xây dựng chương trinh demo cho đồ án môn học của mình. Chương trình về cơ bản hoạt động trơn tru (hầu như không có độ trễ) và có giao diện đẹp mắt dễ hiểu, dễ sử dụng đối với người dùng.

CHƯƠNG 4. KẾT LUẬN

4.1 Kết quả

Mô hình được đánh giá chất lượng trên tập test (tập dữ liệu được tách ra từ tập dữ liệu gốc) thông qua các độ đo cơ bản như accuracy, precision, recall, F1 score. Cả hai mô hình LinearSVC và CatBoost đều cho kết quả tương đối tốt với độ chính xác accuracy 0.83 của LinearSVC và 0.93 của CatBoost. Kết quả chi tiết được trình bày trong hình bên dưới.

	precision	recall	f1-score	support
0	0.82	0.82	0.82	11749
1	0.85	0.85	0.85	14149
accuracy			0.83	25898
macro avg	0.83	0.83	0.83	25898
weighted avg	0.83	0.83	0.83	25898

accuracy_score 0.8348134991119005

Hình 4.1: Kết quả chi tiết của mô hình LinearSVC

	precision	recall	f1-score	support
0	0.92	0.93	0.92	11548
1	0.94	0.93	0.94	14350
accuracy			0.93	25898
macro avg	0.93	0.93	0.93	25898
weighted avg	0.93	0.93	0.93	25898

accuracy_score 0.9315391149895745

Hình 4.2: Kết quả chi tiết của mô hình CatBoost

Với kết quả trên tập test thu được như vậy, có thể nói mô hình đã được huấn luyện thành công vượt qua kết quả mong đợi ban đầu của nhóm là độ chính xác 0.8.

Thêm vào đó, xây dựng thành công chương trình triển khai hai model cũng ghi

nhận là một kết quả tương đối tốt trong quá trình hoàn thiện kiến thức của mình.

4.2 Hướng phát triển

Thông qua kết quả đạt được của hai mô hình, mô hình CatBoost có kết quả tốt hơn mô hình LinearSVC khá nhiều, chúng ta có thể suy lụân rằng tập dữ liệu huấn luyện là tập dữ liệu non-linear, nghĩa là rất khó để tìm được một siêu phẳng có thể phân chia tốt tập dữ liệu thành 2 phần theo cách làm của LinearSVC, mà chúng ta cần phải chuyển tập dữ liệu sang không gian chiều khác theo cách làm của CatBoost thì tại không gian chiều đó chúng ta mới có thể phân chia tốt được tập dữ liệu. Và do đó, hướng phát triển tiếp theo mà chúng em hướng đến là tiếp tục triển khai các mô hình phù hợp với dữ liệu non-linear như CatBoost, có thể là XGBoost, LightGBM với mong muốn thu được kết quả tốt hơn nữa.

Ngoài ra, chúng em sẽ tiếp tục điều chỉnh tham số của hai mô hình hiện tại để cải thiện độ chính xác của chúng.

Một hướng phát triển khác, chúng em sẽ thực hiện các phương pháp tăng cường dữ liệu làm cho bộ dữ liệu lớn hơn, bao quát hơn và mô hình được train trên bộ dữ liệu lớn này cũng sẽ chất lượng hơn.

CHƯƠNG 5. THÔNG TIN KHÁC

5.1 Phân công công việc

Dưới đây là bảng phân công công việc của nhóm. Các thành viên nhiệt tình làm việc nhóm, đóng góp ý kiến và hoàn thành đầy đủ công việc được phân công.

Thành viên	Công việc	Mức độ hoàn thành
Đinh Ngọc Huân	Tiền xử lý dữ liệu Viết báo cáo	100%
Đỗ Minh Hiệp	Triển khai mô hình LinearSVC Viết báo cáo	100%
Chu Đình Đức	Triển khai mô hình CatBoost Viết báo cáo, xây dựng chương trình	100%

Bảng 5.1: Bảng phân công công việc

5.2 Gói phần mềm

Việc tự code lại mô hình CatBoost rất phức tạp và tốn nhiều thời gian cũng như công sức, do đó trong khuôn khổ và thời gian cho phép của đồ án môn học, chúng em đã sử dụng thư viện CatBoost.

Bên cạnh đó, chúng em sử dụng gói phần mềm Streamlit phục vụ cho việc triển khai chương trình.

TÀI LIỆU THAM KHẢO

[1] Liudmila Prokhorenkova et al., "CatBoost: unbiased boosting with categorical features" *Machine Learning*, 2019

[2] Scikit-learn developers, *Support Vector Machines*. [Online]. Available: https://scikit-learn.org/stable/modules/svm.html (visited on 01/08/2023).

[3] Yandex developers, *CatBoost Classifier*. [Online]. Available: https://catboost.ai/en/docs/concepts/python-reference_catboostclassifier (visited on 01/08/2023).

[4] Streamlit Inc., *Streamlit documentation*. [Online]. Available: https://docs.streamlit.io/ (visited on 01/08/2023).

[5] Brain John, *When to Choose CatBoost Over XGBoost or LightGBM*. [Online]. Available: https://neptune.ai/blog/when-to-choose-catboost-over-xgboost-or-lightgbm (visited on 01/08/2023).

[6] Jason Brownlee, *How to Use StandardScaler and MinMaxScaler Transforms in Python*. [Online]. Available: https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/ (visited on 01/08/2023).

[7] Nguyen Chien Thang, *MIAI_XGBoost*. [Online]. Available: https://github.com/thangnch/MIAI_XGBoost (visited on 01/08/2023)

[8] Bui Tien Tung, *Gradient Boosting - Tất tần tật về thuật toán mạnh mẽ nhất trong Machine Learning*. [Online]. Available: https://viblo.asia/p/gradient-boosting-tat-tan-tat-ve-thuat-toan-manh-me-nhat-trong-machine-learning-YWOZrN7vZQ0 (visited on 01/08/2023)

[9] Pham Dinh Khanh, *Phương pháp tăng cường (Boosting)*. [Online]. Available: https://phamdinhkhanh.github.io/deepai-book/ch_ml/index_Boosting.html (visited on 01/08/2023)