ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - CƠ - TIN HỌC

TIỂU LUẬN MÔN HỌC MÁY

Dự đoán tỉ lệ sống sót trên tập dữ liệu Titanic

Người hướng dẫn: TS. Cao Văn Chung

Nhóm sinh viên: Nguyễn Hữu Trung Kiên - 21002152
Lê Quang Đạt - 21002129

Hà Nội - 2024

Mục lục

1 Giới thiệu về học máy

1.1 Khái niệm về học máy

Học máy hay máy học trong tiếng Anh là Machine learning, viết tắt là ML.

Học máy (ML) là một công nghệ phát triển từ lĩnh vực trí tuệ nhân tạo (Artificial Intelligence). Các thuật toán ML là các chương trình máy tính có khả năng học hỏi về cách hoàn thành các nhiệm vụ và cách cải thiện hiệu suất theo thời gian.

ML vẫn đòi hỏi sự đánh giá của con người trong việc tìm hiểu dữ liệu cơ sở và lựa chọn các kĩ thuật phù hợp để phân tích dữ liệu. Đồng thời, trước khi sử dụng, dữ liệu phải sạch, không có sai lệch và không có dữ liệu giả.

Các mô hình ML yêu cầu lượng dữ liệu đủ lớn để "huấn luyện" và đánh giá mô hình. Trước đây, các thuật toán ML thiếu quyền truy cập vào một lượng lớn dữ liệu cần thiết để mô hình hóa các mối quan hệ giữa các dữ liệu. Sự tăng trưởng trong dữ liệu lớn (big data) đã cung cấp các thuật toán ML với đủ dữ liệu để cải thiện độ chính xác của mô hình và dự đoán.

1.2 Các loại thuật toán học máy

Các thuật toán học máy có thể được phân loại thành nhiều nhóm khác nhau dựa trên phương pháp học và loại dữ liệu đầu vào. Dưới đây là các loại thuật toán học máy chính:

- **Học có giám sát:**

  Học có giám sát (Supervised Learning) là một phương pháp học máy trong đó mô hình được huấn luyện dựa trên một tập dữ liệu đã được gán nhãn. Mỗi mẫu trong tập dữ liệu huấn luyện bao gồm một cặp đầu vào và đầu ra mong muốn, và mục tiêu của mô hình là học cách ánh xạ từ đầu vào sang đầu ra. Quá trình này bao gồm việc tìm kiếm các mẫu trong dữ liệu và sau đó sử dụng các mẫu này để dự đoán kết quả cho dữ liệu mới.

  Ứng dụng của kĩ thuật học có giám sát: Xác định tín hiệu hay biến số tốt nhất để dự báo lợi nhuận trong tương lai của cổ phiếu hoặc dự đoán xu hướng thị trường chứng khoán.

- **Học không giám sát:**

  Học không giám sát (Unsupervised Learning) là một phương pháp học máy mà trong đó mô hình được huấn luyện mà không có dữ liệu được gán nhãn trước. Thay vì dựa vào các cặp đầu vào-đầu ra, mô hình học không giám sát nhằm mục đích khám phá cấu trúc ẩn hoặc mẫu trong dữ liệu. Phương pháp này hữu ích khi chúng ta không có sẵn dữ liệu được gán nhãn hoặc muốn tìm hiểu thêm về dữ liệu.

Ứng dụng của học không giám sát: Phân loại các công ty thành các nhóm công ty tương đồng dựa trên đặc điểm của chúng thay vì sử dụng tiêu chuẩn của các nhóm ngành hoặc các quốc gia.

- **Học bán giám sát:**

  Học bán giám sát (Semi-Supervised Learning) là một phương pháp học máy kết hợp giữa học có giám sát (supervised learning) và học không giám sát (unsupervised learning). Trong học bán giám sát, mô hình được huấn luyện bằng cách sử dụng một lượng nhỏ dữ liệu đã được gán nhãn (labelled data) và một lượng lớn dữ liệu chưa được gán nhãn (unlabelled data). Mục tiêu là tận dụng thông tin từ cả dữ liệu có nhãn và không có nhãn để cải thiện hiệu suất của mô hình.

  Học bán giám sát có nhiều ứng dụng trong thực tế, đặc biệt khi việc thu thập dữ liệu có nhãn tốn kém hoặc khó khăn như: Xử lý ngôn ngữ tự nhiên (NLP), thị giác máy tính,...

- **Học tăng cường:**

  Học tăng cường (Reinforcement Learning - RL) là một lĩnh vực của học máy trong đó một agent (tác nhân) học cách hành động trong một môi trường bằng cách thực hiện các hành động và nhận phản hồi dưới dạng phần thưởng hoặc phạt. Mục tiêu của agent là tối đa hóa tổng phần thưởng nhận được qua thời gian. RL khác với học có giám sát ở chỗ nó không học từ một tập dữ liệu huấn luyện cố định mà thông qua sự tương tác liên tục với môi trường.

- **Học sâu:**

  Học sâu (Deep Learning) là một nhánh của ML tập trung vào việc sử dụng các mạng nơ-ron nhân tạo nhiều lớp để mô phỏng hoạt động của não người trong việc xử lý và hiểu dữ liệu phức tạp. Các mô hình học sâu có khả năng học từ dữ liệu có cấu trúc lớn và phức tạp, cho phép chúng tự động học các đặc trưng từ dữ liệu thô mà không cần phải thiết kế thủ công các đặc trưng đó.

1.3 Ứng dụng của học máy

Các thuật toán ML đang được sử dụng để phân tích dữ liệu lớn (big data) để giúp dự đoán xu hướng hoặc sự kiện thị trường, ví dụ như dự đoán kết quả cuộc bầu cử chính trị.

Các thuật toán nhận dạng hình ảnh hiện có thể phân tích dữ liệu từ các hệ thống chụp ảnh vệ tinh để cung cấp thông tin về số lượng khách hàng tại các bãi đậu xe của cửa hàng bán lẻ, hoạt động vận chuyển và cơ sở sản xuất, và sản lượng nông nghiệp...Những thông tin này sẽ cung cấp dữ liệu đầu vào cho các mô hình định giá hoặc các mô hình kinh tế.

2 Giảm chiều dữ liệu

2.1 Khái quát về bộ dữ liệu

Bộ dữ liệu **Titanic** là một trong những bộ dữ liệu phổ biến và được sử dụng nhiều trong lĩnh vực học máy và phân tích dữ liệu. Bộ dữ liệu này được tạo ra để phân loại các hành khách trên tàu Titanic đã sống sót hay đã chết trong thảm họa khi tàu chìm vào năm 1912.

Bộ dữ liệu được chia thành 2 tập:

- Tập huấn luyện: Gồm 12 trường thông tin với 891 bản ghi.
- Tập kiểm thử: Gồm 11 trường thông tin (không có trường Survived) với 418 bản ghi.
- Ngoài ra, còn có 1 tập "gender_submission" lưu các giá trị mục tiêu của tập kiểm thử, phục vụ trong quá trình đánh giá độ chính xác của mô hình sau khi phân loại.

2.1.1 Cấu trúc của bộ dữ liệu Titanic

- **Survived (Sống sót):** Biến mục tiêu, cho biết liệu một hành khách đã sống sót (1) hoặc đã chết (0).
- **Pclass (Hạng ghế):** Loại hạng ghế của hành khách (1, 2 hoặc 3).
- **Name (Tên):** Tên của hành khách.
- **Sex (Giới tính):** Giới tính của hành khách (Nam hoặc Nữ).
- **Age (Tuổi):** Tuổi của hành khách.
- **SibSp (Số lượng anh chị em hoặc vợ/chồng đi cùng trên tàu):** Biến số liên quan đến số lượng anh chị em hoặc vợ/chồng đi cùng trên tàu.
- **Parch (Số lượng cha mẹ hoặc con cái đi cùng trên tàu):** Biến số liên quan đến số lượng cha mẹ hoặc con cái đi cùng trên tàu.
- **Ticket (Vé):** Số vé của hành khách.
- **Fare (Giá vé):** Giá vé mà hành khách đã trả.
- **Cabin (Cabin):** Số cabin của hành khách.
- **Embarked (Cảng lên tàu):** Cảng mà hành khách lên tàu (C = Cherbourg, Q = Queenstown, S = Southampton).

2.1.2 Tiền xử lý dữ liệu

Quan sát tổng thể các trường thông tin của bộ dữ liệu:

```
Training Set
PassengerId column missing values: 0
Survived column missing values: 0
Pclass column missing values: 0
Name column missing values: 0
Sex column missing values: 0
Age column missing values: 177
SibSp column missing values: 0
Parch column missing values: 0
Ticket column missing values: 0
Fare column missing values: 0
Cabin column missing values: 687
Embarked column missing values: 2


Test Set
PassengerId column missing values: 0
Pclass column missing values: 0
Name column missing values: 0
Sex column missing values: 0
Age column missing values: 86
SibSp column missing values: 0
Parch column missing values: 0
Ticket column missing values: 0
Fare column missing values: 1
Cabin column missing values: 327
Embarked column missing values: 0
```

Hình 1: Dữ liệu bị thiếu

Như đã thấy ở trên, 1 vài cột có dữ liệu bị thiếu:

- Tập huấn luyện có dữ liệu bị thiếu ở các cột ‘Age‘, ‘Cabin‘, ‘Embarked‘.
- Tập kiểm thử có dữ liệu bị thiếu ở ‘Age‘, ‘Fare‘và ‘Cabin‘.

‘Age‘: Các dữ liệu bị thiếu của cột ‘Age‘ thường được điền bằng cách lấy giá trị trung bình của các giá trị không thiếu, nhưng sử dụng giá trị trung bình tuổi trên toàn bộ dữ liệu là lựa chọn không tốt, sẽ khôn ngoan hơn nếu gom nhóm theo ‘Pclass‘ và tính giá trị trung bình theo từng nhóm vì ‘Pclass‘ có hệ số tương quan cao so với ‘Age‘

(0.408106). Việc nhóm các độ tuổi theo hạng hành khách cũng hợp lý hơn thay vì các đặc điểm khác.

	Feature 1	Feature 2	Correlation Coefficient
0	Age	Age	1.000000
9	Age	Pclass	0.408106
18	Age	SibSp	0.243699
21	Age	Fare	0.178740
26	Age	Parch	0.150917
30	Age	Survived	0.077221
41	Age	PassengerId	0.028814

Hình 2: Tương quan giữa Age và các cột thông tin khác

- Việc nhóm theo ‘Pclass‘ giúp phản ánh chính xác hơn sự phân bố tuổi trong từng hạng vé khác nhau. Các hành khách ở hạng vé khác nhau có xu hướng có độ tuổi khác nhau. Ví dụ, hành khách ở hạng nhất có thể có độ tuổi trung bình khác với hành khách ở hạng ba.
- Điền giá trị thiếu dựa trên nhóm "Pclass" giúp duy trì phân phối dữ liệu ban đầu. Điều này đặc biệt quan trọng khi thực hiện phân tích thống kê hoặc xây dựng mô hình máy học, vì dữ liệu có phân phối chuẩn xác hơn sẽ giúp mô hình học tốt hơn và phân tích có ý nghĩa hơn.

Để tăng độ chính xác, ‘Sex‘ được sử dụng làm cấp độ thứ hai của việc gom nhóm trong khi điền vào các giá trị ‘Age‘ còn thiếu. Như được thấy từ bên dưới, các nhóm ‘Pclass‘ và ‘Sex‘ có các giá trị ‘Age‘ trung bình riêng biệt. Khi hạng hành khách tăng lên thì độ tuổi trung bình của cả nam và nữ cũng tăng theo. Tuy nhiên, nữ giới có xu hướng có độ tuổi trung bình thấp hơn nam giới một chút. Độ tuổi trung bình bên dưới được sử dụng để điền các giá trị còn thiếu trong cột ‘Age‘.

```
Median age of Pclass 1 females: 36.0
Median age of Pclass 1 males: 42.0
Median age of Pclass 2 females: 28.0
Median age of Pclass 2 males: 29.5
Median age of Pclass 3 females: 22.0
Median age of Pclass 3 males: 25.0
Median age of all passengers: 26.0
```

Hình 3: Độ tuổi trung bình theo Pclass và Sex

‘Embarked‘:là một đặc trưng phân loại và chỉ thiếu 2 giá trị trong toàn bộ tập dữ liệu. Cả hai hành khách đó đều là nữ, thuộc tầng lớp thượng lưu và có cùng số vé. Điều này có nghĩa là họ biết nhau và cùng lên đường từ cùng một cảng.

	Age	Cabin	Embarked	Fare	Name	Parch	PassengerId	Pclass	Sex	SibSp	Survived	Ticket
61	38.0	B28	NaN	80.0	Icard, Miss. Amelie	0	62	1	female	0	1.0	113572
829	62.0	B28	NaN	80.0	Stone, Mrs. George Nelson (Martha Evelyn)	0	830	1	female	0	1.0	113572

Hình 4: giá trị Embarked bị thiếu

Thực tế, giá trị ‘Embarked‘ này thường không cần thiết cho mục tiêu phân loại ‘Survived‘, hoặc chỉ đơn giản là bỏ đi, không cần xử lý, hoặc có thể gán 1 giá trị bất kì cho 2 vị trí bị thiếu này. Ở đây, nhóm điền giá trị "C" cho 2 giá trị còn thiếu này.

‘Fare‘: Chỉ có một hành khách bị thiếu Giá trị ‘Fare‘. Có thể giả định rằng ‘Fare‘ có liên quan đến quy mô nhóm (‘Parch‘ và ‘SibSp‘) và các tính năng của ‘Pclass‘.

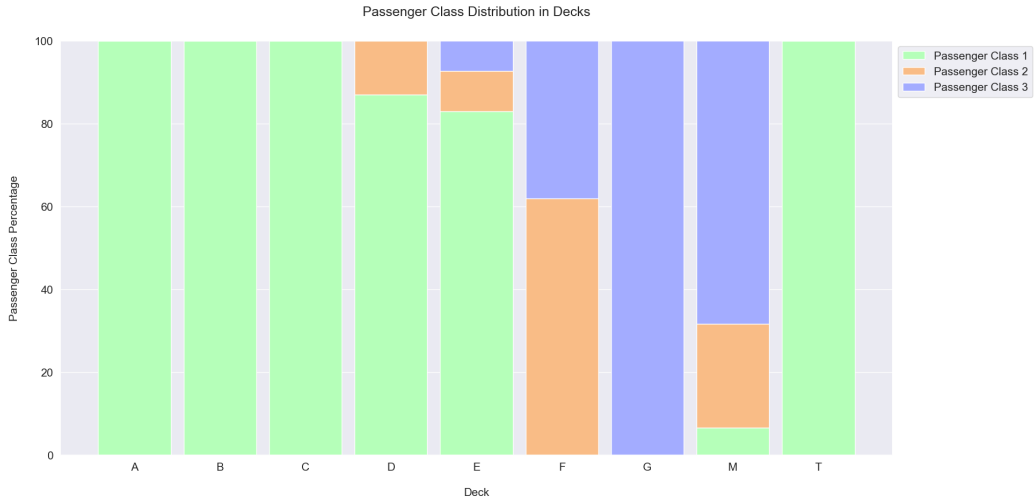
	Age	Cabin	Embarked	Fare	Name	Parch	PassengerId	Pclass	Sex	SibSp	Survived	Ticket
1043	60.5	NaN	S	NaN	Storey, Mr. Thomas	0	1044	3	male	0	NaN	3701

Hình 5: giá trị Fare bị thiếu

Giá trị giá vé trung bình của một nam giới có vé hạng ba và không có gia đình là một lựa chọn hợp lý để lấp đầy giá trị còn thiếu.

‘Cabin‘: hơi phức tạp. Phần lớn ‘Cabin‘ bị thiếu và không thể bỏ qua hoàn toàn đặc trưng này vì một số cabin có thể có tỷ lệ sống sót cao hơn. Những chữ cái đầu tiên của giá trị Cabin là các tầng chứa các cabin. Những boong đó chủ yếu được tách ra cho một hạng hành khách, nhưng một số trong số chúng được sử dụng bởi nhiều hạng hành khách.

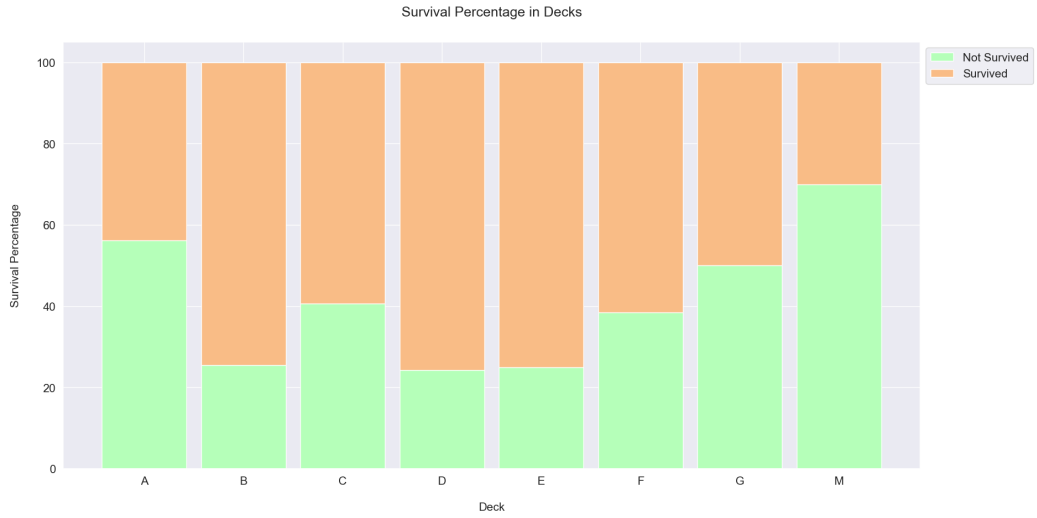
- Boong A, B, C và T chỉ dành cho hành khách hạng 1.
- Boong D và E dành cho mọi tầng lớp.
- Boong F và G dành cho cả hành khách hạng 2 và hạng 3.
- Từ A đến G, khoảng cách đến cầu thang tăng lên có thể là yếu tố sống còn.



Hình 6: Pclass với Cabin

- 100% boong A, B, C là hành khách hạng 1.
- Tầng D có 87% hành khách hạng 1 và 13% hành khách hạng 2.
- Tầng E có 83% hành khách hạng 1, 10% hành khách hạng 2 và 7% hành khách hạng 3.
- Tầng F có 62% hành khách hạng 2 và 38% hành khách hạng 3.
- 100% khoang G là hành khách hạng 3.
- Có một người trên boong thuyền ở cabin T và anh ta là hành khách hạng nhất. Hành khách ở khoang T có tương đồng nhất với hành khách ở khoang A nên được xếp vào nhóm A.
- Hành khách được gắn nhãn M là giá trị còn thiếu trong ‘Cabin‘.

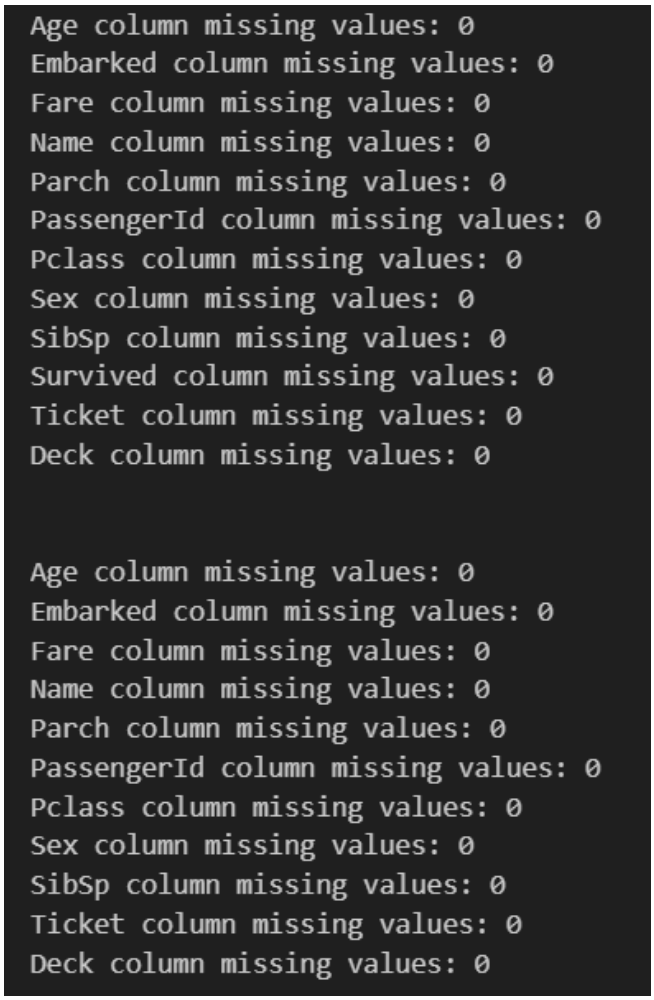
Mỗi boong đều có tỷ lệ sống sót khác nhau và thông tin đó không thể bị loại bỏ. Boong B, C, D và E có tỷ lệ sống sót cao nhất. Những boong đó chủ yếu dành cho hành khách hạng nhất. M có tỷ lệ sống sót thấp nhất, chủ yếu là hành khách hạng 2 và hạng 3. Tóm lại, cabin được hành khách hạng 1 sử dụng có tỷ lệ sống sót cao hơn cabin được hành khách hạng 2 và 3 sử dụng. Boong M (Missing Cabin value) có tỷ lệ sống sót thấp nhất vì không lấy được dữ liệu cabin của nạn nhân. Đó là lý do tại sao, việc gắn nhãn nhóm đó là M là cách hợp lý để xử lý dữ liệu còn thiếu.



Hình 7: Survived với Cabin

- Boong A, B và C được dán nhãn ABC vì tất cả đều chỉ có hành khách hạng 1.
- Boong D và E được dán nhãn là DE vì cả hai đều có sự phân bố hạng hành khách giống nhau và tỷ lệ sống sót như nhau.
- Boong F và G được dán nhãn là FG vì lý do tương tự ở trên.
- Boong M không cần phải nhóm với các bộ bài khác vì nó rất khác biệt với những bộ bài khác và có tỷ lệ sống sót thấp nhất.

Sau khi điền các giá trị còn thiếu trong ‘Age‘, ‘Embarked‘, ‘Fare‘ và ‘Cabin‘, không còn giá trị nào còn thiếu trong cả tập huấn luyện và tập kiểm tra. ‘Cabin‘ bị loại bỏ và thay thế bằng ‘Deck‘.



Hình 8: Không còn giá trị bị thiếu

2.2 Giảm chiều dữ liệu

2.2.1 Khái niệm

PCA (Principal Component Analysis) là một phương pháp thống kê dùng để giảm chiều dữ liệu trong các tập dữ liệu lớn mà vẫn giữ lại được càng nhiều thông tin quan trọng càng tốt. Mục đích là biến đổi một tập dữ liệu có nhiều chiều thành một tập dữ liệu có ít chiều hơn, nhưng vẫn giữ lại được hầu hết các đặc trưng quan trọng của dữ liệu gốc. Điều này giúp giảm độ phức tạp của dữ liệu, dễ dàng hơn trong việc trực quan hóa và xử lý.

Ưu điểm:

- Giảm chiều dữ liệu: Giúp giảm số lượng biến đầu vào, từ đó giảm chi phí tính toán và tránh overfitting.
- Trực quan hóa dữ liệu: Giúp trực quan hóa dữ liệu đa chiều trong không gian 2D

hoặc 3D.

- Loại bỏ nhiễu: Giảm ảnh hưởng của nhiễu trong dữ liệu bằng cách giữ lại các thành phần chính có phương sai lớn.

Nhược điểm:

- Mất thông tin: Quá trình giảm chiều có thể làm mất một số thông tin, đặc biệt là khi chỉ giữ lại một số lượng nhỏ các thành phần chính.
- Không dễ giải thích: Các thành phần chính mới không dễ giải thích theo các biến gốc, do chúng là tổ hợp tuyến tính của các biến gốc.

2.2.2 Các bước giảm chiều

- Tính vector kỳ vọng của toàn bộ dữ liệu:

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

- Trừ mỗi điểm dữ liệu đi vector kỳ vọng của toàn bộ dữ liệu:

$$\hat{x}_n = x_n - \bar{x}$$

- Tính ma trận hiệp phương sai:

$$S = \frac{1}{N} X X^T$$

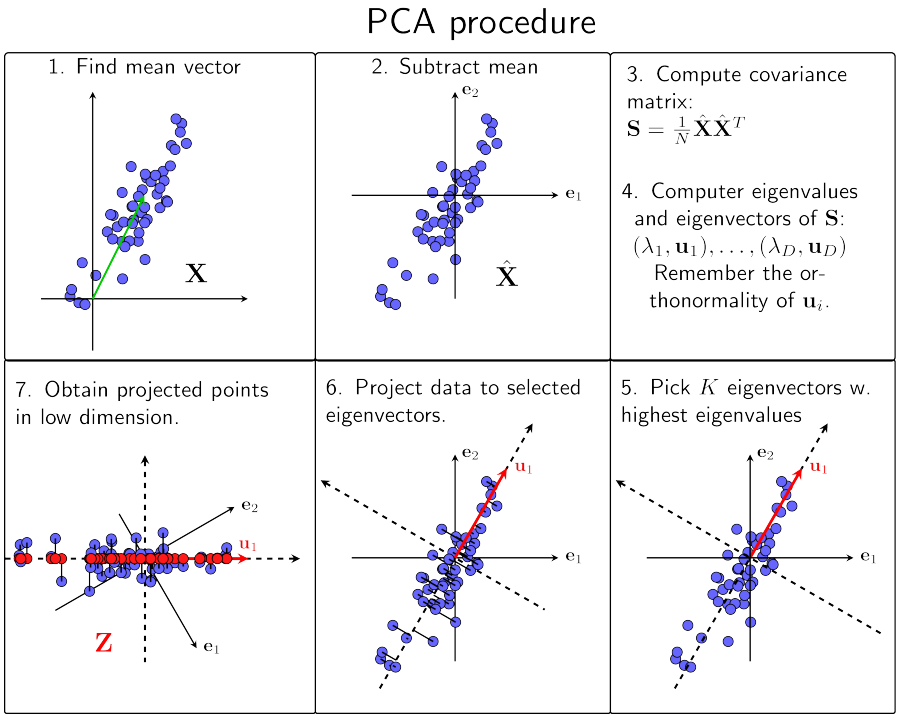
- Tính các trị riêng và vector riêng có norm bằng 1 của ma trận này, sắp xếp chúng theo thứ tự giảm dần của trị riêng.
- Chọn K vector riêng ứng với K giá trị riêng lớn nhất để xây dựng ma trận U_K có các cột tạo thành một hệ trực giao. K vectors này, còn được gọi là các thành phần chính, tạo thành một không gian con gần với phân bố của dữ liệu ban đầu đã chuẩn hoá.
- Chiếu dữ liệu ban đầu đã chuẩn hoá $\hat{X}$ xuống không gian con tìm được.
- Dữ liệu mới chính là toạ độ của các điểm dữ liệu trên không gian mới

$$Z = U_K^T X$$

Dữ liệu ban đầu có thể tính được xấp xỉ theo dữ liệu mới như sau:

$$x \approx U_K Z + \bar{x}$$

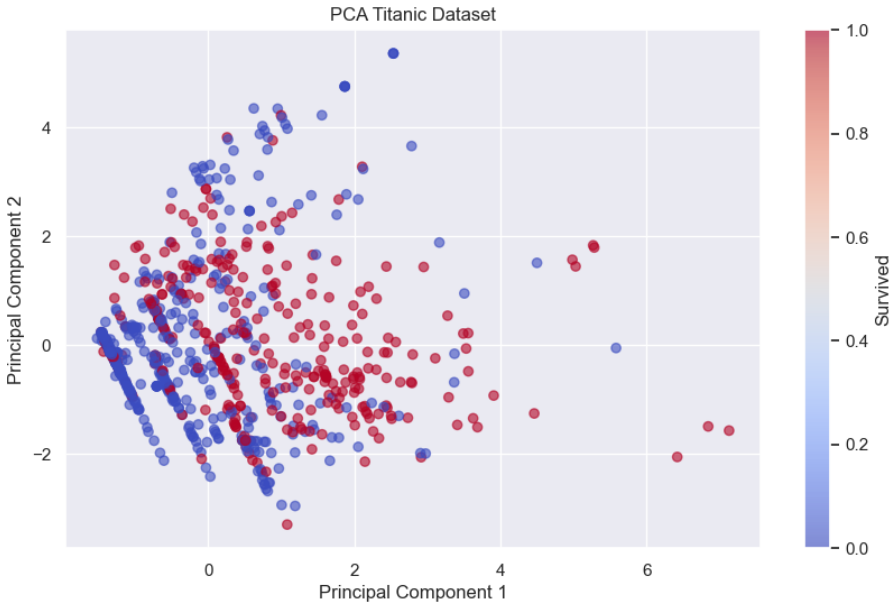
Các bước thực hiện có thể xem hình dưới đây:



Hình 9: Các bước PCA

2.2.3 Kết quả trực quan

Hiển thị trực quan kết quả giảm chiều dữ liệu:



3 Phân cụm dữ liệu

3.1 Xác định cụm tối ưu qua phương pháp Elbow

Phương pháp Elbow là một cách giúp chúng ta lựa chọn được số lượng các cụm phù hợp dựa vào đồ thị trực quan hóa bằng cách nhìn vào sự suy giảm của hàm biến dạng và lựa chọn ra điểm khuỷu tay (elbow point).

Nó chọn ra phạm vị các giá trị và chọn ra giá trị tốt nhất trong số đó. Nó tính tổng bình phương của các điểm và tính khoảng cách trung bình.

Các bước của phương pháp elbow

1. Lựa chọn số lượng cụm K để thử nghiệm, tại đây k được lấy thuộc khoảng từ 2 đến 11.
2. Huấn luyện mô hình với mỗi giá trị k. Sử dụng mô hình phân loại với giá trị k tương ứng và đánh giá hiệu suất của mô hình.
3. Đo lường độ tốt của mô hình. Sử dụng các phép đo như SSD (Sum of Squared Distances) cho K-means hoặc các phép đo khác phù hợp cho mô hình

3.2 Phân cụm dữ liệu sử dụng k-means

Thuật toán K-Means Clustering là một phương pháp phân cụm dữ liệu phổ biến trong học máy, được sử dụng để nhóm các điểm dữ liệu có đặc điểm tương đồng vào các cụm riêng biệt.

Trong đề tài này, K-Means Clustering sẽ được áp dụng cho tập dữ liệu Titanic để phân tích các yếu tố ảnh hưởng đến tỷ lệ sống sót của hành khách trên con tàu Titanic.

Các bước trong thuật toán k-means:

- **Khởi tạo tâm cụm:** Thuật toán bắt đầu bằng việc lựa chọn ngẫu nhiên k điểm dữ liệu (cụm) trong tập dữ liệu.
- **Phân cụm:** Sau khi có k số cụm ban đầu, chúng ta sẽ tính toán khoảng cách giữa từng điểm dữ liệu với k cụm này và gán điểm dữ liệu đó vào cụm gần nó nhất. Khoảng cách giữa 2 điểm dữ liệu được tính bằng khoảng cách Euclidean. Công thức tính khoảng cách Euclidean được biểu diễn như sau.

$$\sqrt{(y_2 - y_1)^2 - (x_2 - X_1)^2}$$

- **Cập nhật lại tâm cụm:** Tính toán lại tâm cụm bằng cách lấy trung bình của các điểm dữ liệu trong mỗi cụm. Công thức tính tâm cụm mới

$$c_j = \frac{1}{N_j} \sum_{x_i \in C_j} x_i$$

Trong đó N_j là số điểm dữ liệu trong cụm j và C_j là tập hợp các điểm dữ liệu thuộc cụm j.

- Lặp lại các bước sau đến khi hội tụ:
  1. Bước 1: Với mỗi điểm dữ liệu, gán điểm dữ liệu đó vào cụm có khoảng cách đến tâm cụm của cụm là nhỏ nhất.
  2. Bước 2: Với mỗi cụm, xác định lại tâm cụm của tất cả các điểm dữ liệu được gán vào cụm đó.
- Điều kiện hội tụ (điều kiện dừng thuật toán) theo một số cách như sau:
  1. Tại 1 vòng lặp: có ít các điểm dữ liệu được gán sang cụm khác
  2. Tâm cụm (centroid) không thay đổi nhiều
  3. Giá trị hàm mất mát không thay đổi nhiều

3.3 Phân cụm dữ liệu với gaussian mixture models

Gaussian Mixture Models (GMM) là một phương pháp thống kê mạnh mẽ được sử dụng để phân cụm và mô hình hóa phân phối dữ liệu. GMM được xây dựng trên cơ sở lý thuyết rằng tất cả dữ liệu đều được tạo ra từ một tổ hợp (mixture) của nhiều phân phối Gaussian (chuẩn). Khác với các phương pháp phân cụm đơn giản như K-Means, GMM cho phép mỗi cụm có hình dạng elip và không yêu cầu các cụm có kích thước và hình dạng giống nhau.

Gaussian mixture models bao gồm nhiều thành phần cơ bản:

- Các phân phối thành phần (Component Distributions): Mỗi phân phối Gaussian trong GMM được gọi là một thành phần. Mỗi thành phần mô hình hóa một cụm hoặc nhóm dữ liệu trong tổng thể dữ liệu.
- Trung bình (Means): Trung bình μ của mỗi phân phối thành phần cho biết vị trí trung tâm của cụm dữ liệu đó trong không gian nhiều chiều. Trung bình là một yếu tố quan trọng giúp xác định vị trí của từng cụm.
- Phương sai (Variances): Phương sai σ^2 của mỗi thành phần thể hiện mức độ phân tán của dữ liệu trong cụm đó. Phương sai có thể khác nhau giữa các thành phần, cho phép mỗi cụm có hình dạng và kích thước riêng biệt.
- Hệ số hỗn hợp (Mixture Coefficients): Hệ số hỗn hợp π đại diện cho trọng số của mỗi phân phối Gaussian trong tổng thể GMM, cho biết tầm quan trọng tương đối của từng phân phối thành phần. Tổng các hệ số hỗn hợp bằng 1, thể hiện việc chia tỉ lệ đóng góp của mỗi thành phần vào mô hình tổng thể.

Sơ bộ về thuật toán EM (Expectation-Maximization):

1. Bước E (Expectation Step): Ở bước này, dữ liệu có sẵn được sử dụng để dự đoán giá trị của các biến còn thiếu.

2. Bước M (Maximization Step): Dựa trên các giá trị ước tính được tạo ra trong bước E, dữ liệu hoàn chỉnh được sử dụng để cập nhật các tham số.

4 Mô hình phân loại

4.1 Random Forest

4.1.1 Khái niệm

Cây quyết định là một mô hình khá nổi tiếng hoạt động trên cả hai lớp bài toán phân loại và dự báo của học có giám sát. Ý tưởng chính của mô hình là xây dựng một đồ thị dạng câu hỏi để đưa ra dự báo.

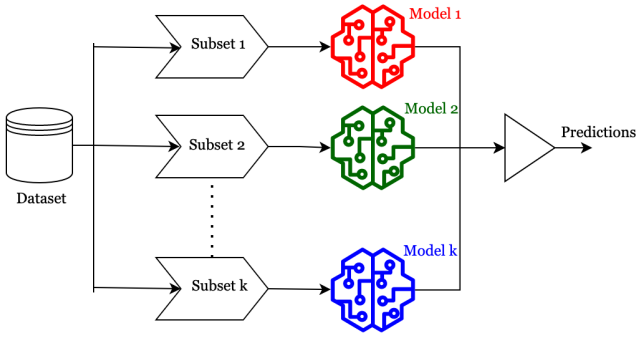
Dù có độ chính xác khá cao nhưng cây quyết định tồn tại những hạn chế lớn đó là:

- Dễ xảy ra quá khớp nếu số lượng các đặc trưng để hỏi lớn. Khi độ sâu của cây quyết định không bị giới hạn thì có thể tạo ra những node lá chỉ có một vài quan sát. Những kết luận dự báo từ chúng thường chỉ đúng trên tập huấn luyện mà không đúng trên tập kiểm tra.
- Trong tình huống bộ dữ liệu có số lượng biến lớn. Một cây quyết định có độ sâu giới hạn (để giảm thiểu quá khớp) thường bỏ sót những biến quan trọng.
- Cây quyết định chỉ tạo ra một kịch bản dự báo duy nhất cho mỗi một quan sát nên nếu model có hiệu suất kém thì kết quả sẽ bị chệch.

Nếu như sức mạnh của một cây quyết định là yếu thì hợp sức của nhiều cây quyết định sẽ trở nên mạnh mẽ hơn. Ý tưởng của sự hợp sức đã hình thành nên mô hình rừng cây (Random Forest).

Mô hình rừng cây được huấn luyện dựa trên sự phối hợp giữa luật kết hợp (ensembling) và quá trình lấy mẫu tái lặp (boostrapping). Cụ thể thuật toán này tạo ra nhiều cây quyết định mà mỗi cây quyết định được huấn luyện dựa trên nhiều mẫu con khác nhau và kết quả dự báo là bầu cử (voting) từ toàn bộ những cây quyết định. Như vậy một kết quả dự báo được tổng hợp từ nhiều mô hình nên kết quả của chúng sẽ không bị chệch. Đồng thời kết hợp kết quả dự báo từ nhiều mô hình sẽ có phương sai nhỏ hơn so với chỉ một mô hình. Điều này giúp cho mô hình khắc phục được hiện tượng quá khớp.

4.1.2 Mô hình kết hợp (ensemble model)



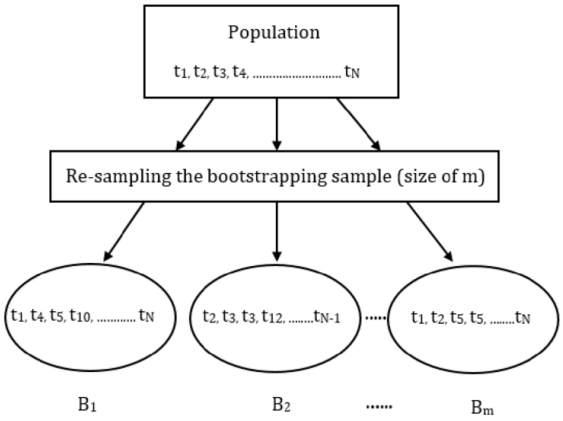
Hình 10: Mô hình ensemble

Giả định rằng bạn đang xây dựng mô hình phân loại nhị phân ảnh chó và mèo lần lượt tương ứng với hai nhãn 0 và 1. Với một hình ảnh cụ thể, nếu chỉ sử dụng một mô hình duy nhất thì kết quả dự báo trả về có xác suất thuộc về nhãn mèo chỉ là 0.6. Đây là một xác suất không quá cao nên bạn không chắc chắn hình ảnh của của mình là mèo.

Bởi vì không chắc chắn, sẽ muốn tham vấn kết quả từ nhiều mô hình hơn. Chính vì vậy, quyết định xây dựng 9 mô hình khác nhau và tiến hành bầu cử kết quả trả về giữa chúng. Do đây là một trường hợp khó phát hiện, chẳng hạn ảnh bị nhoè và con vật đang núp dưới một gốc cây nên các mô hình đều dự báo xác suất không quá gần 1. Nhưng bất ngờ đó là trong kết quả trả về từ 9 mô hình thì có 8 mô hình dự báo nhãn 1 và 1 mô hình dự báo nhãn 0. Như vậy căn cứ vào kết quả bầu cử bạn có thể tin cậy rằng nhãn dự báo cho bức ảnh là mèo là đúng.

4.1.3 Lấy mẫu tái lập (bootstrapping)

Giả định dữ liệu huấn luyện mô hình là một tập $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ bao gồm N quan sát. Thuật toán rừng cây sẽ sử dụng phương pháp lấy mẫu tái lập để tạo thành B tập dữ liệu con. Quá trình lấy mẫu tái lập này còn gọi là bỏ túi (bagging). Tức là sẽ thực hiện M lượt nhặt các mẫu từ tổng thể và bỏ vào túi để tạo thành tập $B_i = \{(x_1^i, y_1^i), (x_2^i, y_2^i), ..., (x_M^i, y_M^i)\}$. Tập B_i cho phép các phần tử được lặp lại. Như vậy sẽ tồn tại những quan sát thuộc D nhưng không thuộc B_i. Đây là những quan sát chưa được bỏ vào túi và chúng ta gọi chúng là nằm ngoài túi (out of bag).



Hình 11: Lấy mẫu bootstrap

Với mỗi tập dữ liệu B_i xây dựng một mô hình cây quyết định và trả về kết quả dự báo là $\hat{y_j}^i = f_i(x_j)$. Trong đó $\hat{y_j}^i$ là dự báo của quan sát thứ j từ mô hình thứ i, x_j là giá trị vecto đầu vào, $f_i(.)$ là hàm dự báo của mô hình thứ i. Mô hình dự báo từ cây quyết định là giá trị trung bình hoặc bầu cử của B cây quyết định.

- Đối với mô hình dự báo: Tính giá trị trung bình của các dự báo từ mô hình con.

$$\hat{y_j} = \frac{1}{B}\sum_{n=1}^{B} \hat{y_j}^i$$

- Đối với mô hình phân loại: Thực hiện bầu cử từ các mô hình con để chọn ra nhãn dự báo có tần suất lớn nhất.

$$\hat{y_j} = \arg\max_c \sum_{n=1}^{B} p(\hat{y_j}^i = c)$$

Như vậy phương sai của mô hình trong trường hợp đối với bài toán dự báo:

$$\sigma_{\hat{y}}^2 = \mathrm{Var}\left(\frac{1}{B}\sum_{n=1}^{B} \hat{y}^i\right)$$

$$= \frac{1}{B^2}[\sum_{n=1}^{B} \mathrm{Var}(\hat{y}^i) + 2\sum_{1\le m<n\le B} \mathrm{cov}(y^m, y^n)]$$

Do kết quả của mô hình con A không chịu ảnh hưởng hoặc phụ thuộc vào mô hình con B nên ta có thể giả định kết quả dự báo từ các mô hình là hoàn toàn độc lập nhau. Tức là ta có $\mathrm{cov}(y^m, y^n) = 0, \forall 1 \le m < n \le B$. Đồng thời giả định chất lượng các mô hình

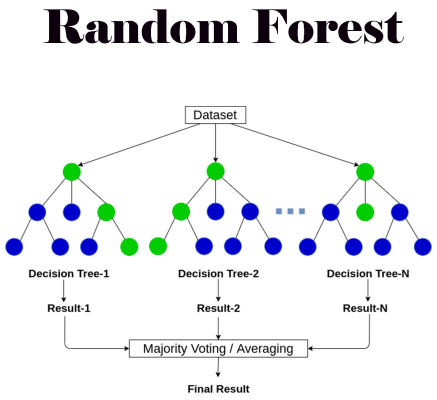
là đồng đều, được thể hiện qua phương sai dự báo là đồng nhất $\text{Var}(\hat{y}^i) = \sigma^2, \forall i \in [1, B]$. Từ đó suy ra:

$$\sigma^2_{\hat{y}} = \frac{1}{B^2} \sum_{n=1}^{B} \text{Var}(\hat{y}^i)$$

$$= \frac{1}{B^2} B_{\sigma^2} = \frac{1}{B} \sigma^2$$

Như vậy nếu sử dụng dự báo là trung bình kết hợp từ nhiều mô hình cây quyết định thì phương sai có thể giảm B lần so với chỉ sử dụng một mô hình duy nhất. Trong một mô hình rừng cây, số lượng các cây quyết định là rất lớn. Do đó phương sai dự báo từ mô hình có thể giảm gấp nhiều lần và tạo ra một dự báo ổn định hơn.

4.1.4 Quá trình huấn luyện



Hình 12: Mô hình rừng ngẫu nhiên

Mô hình rừng cây sẽ áp dụng cả hai phương pháp học kết hợp (ensemble learning) và lấy mẫu tái lập (boostrapping).

- Lấy mẫu tái lập một cách ngẫu nhiên từ tập huấn luyện để tạo thành một tập dữ liệu con.

  Việc lấy mẫu bootstrap trong Random Forest bỏ qua một phần dữ liệu gốc trong mỗi lần lấy mẫu để tạo ra các mẫu bootstrap. Phần dữ liệu bị bỏ qua này gọi là "out-of-bag" (OOB) samples. Trung bình, khoảng 36.8% dữ liệu gốc sẽ không được chọn trong mỗi mẫu bootstrap.

  – Tính toán lý thuyết: Như đã đề cập, xác suất để một mẫu cụ thể không được chọn ít nhất một lần trong quá trình bootstrap (lấy mẫu lại với thay thế) là:

$$(1 - \frac{1}{n})^n$$

Khi n rất lớn, biểu thức này tiến đến $e^{-1} \approx 0.3679$, tức là khoảng 36.8% dữ liệu gốc sẽ là out-of-bag samples.

– Kiểm tra thực nghiệm:

```
Percentage of OOB samples in a single bootstrap sample: 35.47%
Average percentage of OOB samples over 1000 trials: 36.76%
```

Hình 13: Tỉ lệ mẫu out-of-bag

Số mẫu out-of-bag chiếm khoảng 35.47%

- Lựa chọn ra ngẫu nhiên d biến và xây dựng mô hình cây quyết định dựa trên những biến này và tập dữ liệu con ở bước 1. Chúng ta sẽ xây dựng nhiều cây quyết định nên bước 1 và 2 sẽ lặp lại nhiều lần.

  Mẫu bootstrap sẽ được xây dựng dựa trên những đặc trưng được lựa chọn :

  – **'Age':** Hiển nhiên, đặc trưng tuổi tác sẽ ảnh hưởng đến khả năng sống sót. Những người càng cao tuổi thì tỉ lệ sống sót càng thấp đi.

  – **'Pclass':** Những người có hạng vé cao hơn thường được ưu tiên, nên khả năng sống sót cũng cao hơn.

  – **'Sex':** Nữ giới có tỉ lệ sống sót cao hơn nam giới.

  – **'Deck':** Đặc trưng này 1 phần có thể được quyết định bởi 'Pclass', nhưng ngoại trừ hành khách hạng 1 ở riêng biệt, thì có những khoang ở chung hành khách hạng 2 và 3. Những khoang càng xa cầu thang thì tỉ lệ sống sót càng thấp. Việc lựa chọn đặc trưng này sẽ rạch ròi kĩ lưỡng hơn.

  – **'Sibsp' và 'Parch':** Số lượng anh/chị/em hoặc cha mẹ/con cái cũng ảnh hưởng đến khả năng sống sót của hành khách.

- Thực hiện bầu cử hoặc lấy trung bình giữa các cây quyết định để đưa ra dự báo.

4.2 K-Nearest Neighbors

4.2.1 Khái niệm

Trong học máy, K-Nearest Neighbors hay kNN là thuật toán đơn giản nhất trong tất cả các thuật toán học máy. Nó là một thuật toán phi tham số được sử dụng cho các nhiệm vụ phân loại và hồi quy. Phi tham số có nghĩa là không cần có giả định cho việc phân phối dữ liệu. Một vài tham số như "Trọng số của các hàng xóm" (một số cài đặt kNN cho phép các hàng xóm gần hơn có trọng số cao hơn trong việc tính toán dự đoán so với các hàng xóm xa hơn) không phải là bắt buộc và thường được điều chỉnh để tối ưu hóa hiệu suất của mô hình kNN trên dữ liệu cụ thể. Do đó, chúng có thể được gọi là "phi tham số" của thuật toán kNN, bởi vì chúng không cố định mà phụ thuộc vào bài toán và dữ liệu cụ thể mà mô hình đang giải quyết.

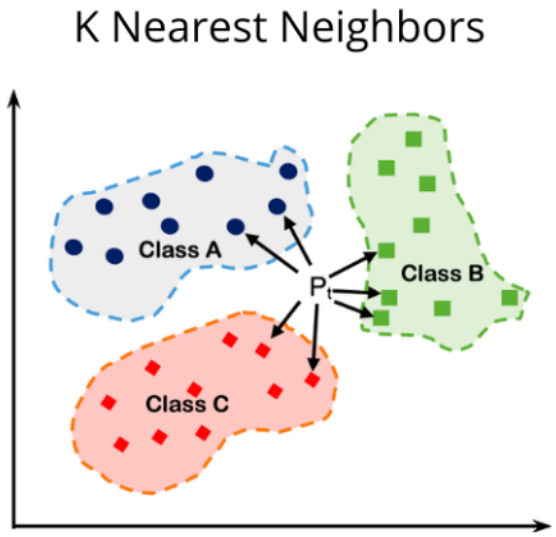
- Trong phân loại, điểm dữ liệu đã cho được phân loại dựa trên phần lớn loại điểm lân cận của nó. Điểm dữ liệu được gán cho lớp thường xuyên nhất trong số k lân cận gần nhất của nó. Thông thường k là số nguyên dương nhỏ. Nếu k=1 thì điểm dữ liệu chỉ được gán cho lớp của điểm lân cận gần nhất đó.
- Trong hồi quy, đầu ra chỉ đơn giản là một số giá trị thuộc tính cho đối tượng. Giá trị này là giá trị trung bình của các giá trị k lân cận gần nhất.

kNN là một kiểu học dựa trên cá thể hoặc học lười biếng. Học lười có nghĩa là nó không yêu cầu bất kỳ điểm dữ liệu huấn luyện nào để tạo mô hình. Tất cả dữ liệu đào tạo sẽ được sử dụng trong giai đoạn thử nghiệm. Điều này làm cho việc đào tạo nhanh hơn và việc kiểm tra chậm hơn và tốn kém hơn. Vì vậy, giai đoạn thử nghiệm đòi hỏi nhiều thời gian và tài nguyên bộ nhớ hơn.

Để đưa ra phân loai, kNN sẽ tính toán khoảng cách của điểm dữ liệu mẫu với toàn bộ các điểm dữ liệu của tập huấn luyện. Với bộ dữ liệu lớn thì việc tính toán sẽ rất mất thời gian. Vậy nên, kNN là lựa chọn phù hợp với bộ dữ liệu Titanic vì kích thước tập huấn luyện không quá lớn.

4.2.2 Nguyên lý hoạt động

Cách thức của thuật toán kNN rất đơn giản. Nó chỉ đơn giản tính toán khoảng cách giữa một điểm dữ liệu mẫu và tất cả các điểm dữ liệu huấn luyện khác. Khoảng cách có thể là khoảng cách Euclide (chủ yếu), khoảng cách Manhattan hoặc khoảng cách Minkowski. Sau đó, nó chọn k điểm dữ liệu gần nhất trong đó k có thể là số nguyên bất kỳ. Cuối cùng, nó gán điểm dữ liệu mẫu cho lớp mà phần lớn k điểm dữ liệu thuộc về.



Hình 14: Mô hình kNN

Khoảng cách Euclid: là một cách đo lường khoảng cách giữa hai điểm trong không

gian Euclid, và nó là một khái niệm cơ bản trong hình học. Trong không gian hai chiều, khoảng cách Euclid giữa hai điểm (x_1, y_1) và (x_2, y_2) được tính bằng công thức:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Khoảng cách Euclid tổng quát giữa hai điểm $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ và $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ trong không gian n chiều được tính bằng công thức:

$$d = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Hãy thực hiện 1 ví dụ nhỏ để hiểu rõ hơn. Giả sử chúng ta có một tập dữ liệu có hai biến được phân loại là Đỏ và Xanh.

Trong thuật toán kNN, k là số lân cận gần nhất. Thông thường, k là số lẻ vì nó giúp quyết định số đông trong lớp. Khi k=1 thì thuật toán được gọi là thuật toán lân cận gần nhất.

Bây giờ, muốn phân loại điểm dữ liệu X mới thành lớp Xanh hoặc lớp Đỏ. Giả sử giá trị của k là 3. Thuật toán kNN bắt đầu bằng cách tính khoảng cách giữa X và tất cả các điểm dữ liệu khác. Sau đó, nó tìm thấy 3 điểm gần nhất có khoảng cách nhỏ nhất tới điểm X.

Ở bước cuối cùng của thuật toán kNN, gán điểm dữ liệu X mới cho phần lớn lớp của 3 điểm gần nhất. Nếu 2 trong 3 điểm gần nhất thuộc lớp Đỏ trong khi 1 điểm thuộc lớp Xanh lam thì phân loại điểm dữ liệu mới là Đỏ.

Tóm lại, các bước thực hiện của kNN như sau:

- Chọn giá trị tối ưu của K.
- Tính khoảng cách (thường là khoảng cách Euclid).
- Tìm hàng xóm gần nhất.
- Bỏ phiếu phân loại hoặc lấy trung bình để hồi quy.

4.2.3 Làm cách nào để xác định số hàng xóm cần thiết

Trong khi xây dựng mô hình phân loại kNN, một câu hỏi lớn là giá trị của các lân cận gần nhất (k) sẽ mang lại độ chính xác cao nhất là bao nhiêu. Đây là một câu hỏi rất quan trọng vì độ chính xác của việc phân loại phụ thuộc vào sự lựa chọn k.

Số lượng láng giềng (k) trong kNN là tham số cần lựa chọn khi xây dựng mô hình. Việc chọn giá trị tối ưu của k trong kNN là bài toán quan trọng nhất. Giá trị k nhỏ có nghĩa là nhiễu sẽ có ảnh hưởng lớn hơn đến kết quả. Vì vậy, khả năng overfit là rất cao.

Giá trị lớn của k làm cho việc tính toán tốn kém về mặt thời gian để xây dựng mô hình kNN. Ngoài ra, giá trị k lớn sẽ có ranh giới quyết định mượt mà hơn, nghĩa là phương sai thấp hơn nhưng độ lệch cao hơn.

Các nhà khoa học dữ liệu chọn giá trị k lẻ nếu số lớp là số chẵn. Để tối ưu hóa kết quả, có thể sử dụng kỹ thuật "Xác thực chéo" để kiểm tra thuật toán kNN với các giá trị khác nhau của k. Mô hình cho độ chính xác tốt có thể coi là sự lựa chọn tối ưu. Nó phụ thuộc vào từng trường hợp riêng lẻ và đôi khi quy trình tốt nhất là chạy qua từng giá trị có thể có của k và kiểm tra kết quả.

Kỹ thuật xác thực chéo (cross-validation): là một phương pháp phổ biến và quan trọng trong machine learning để đánh giá hiệu suất của mô hình.Trong báo cáo này, nhóm sẽ áp dụng kỹ thuật Xác thực chéo k-Fold để đánh giá rõ hơn hiệu suất mô hình. Nó ổn định và kỹ lưỡng hơn so với việc sử dụng phân tách thử nghiệm đào tạo để đánh giá hiệu suất mô hình.

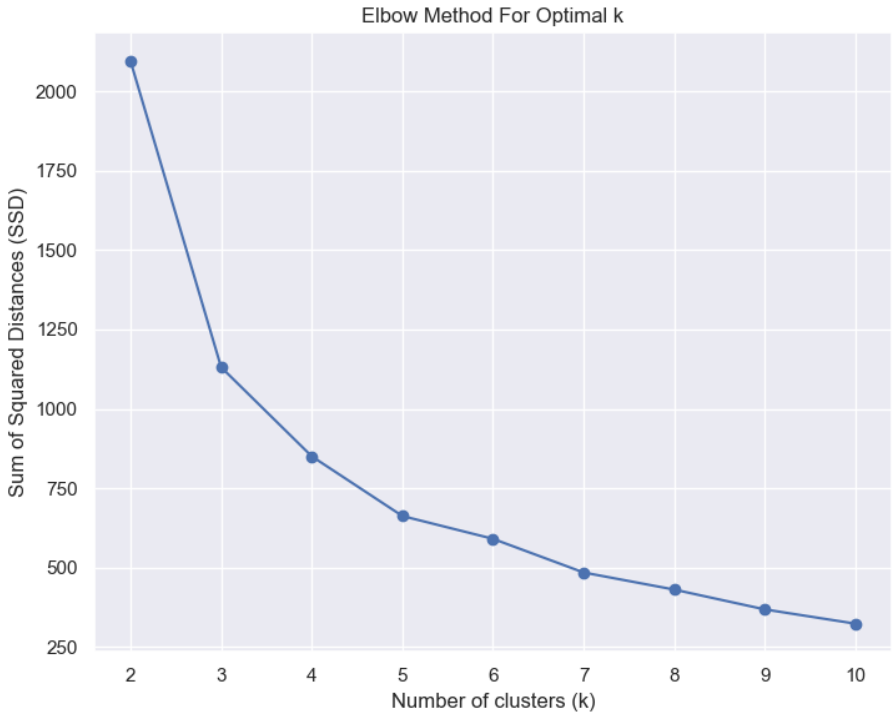
Các bước thực hiện xác thực chéo K-fold:

- Chia dữ liệu: Chia tập dữ liệu thành K folds (phần), mỗi fold có số lượng mẫu gần như bằng nhau.
- Lặp lại quá trình:
  - Đối với mỗi lần lặp, chọn một fold làm tập kiểm tra và các fold còn lại làm tập huấn luyện.
  - Huấn luyện mô hình trên tập huấn luyện và đánh giá hiệu suất trên tập kiểm tra.
- Tính toán hiệu suất: Đo lường và ghi nhận kết quả đánh giá của mô hình sau mỗi lần thử nghiệm.

5 Kết quả và đánh giá

5.1 Trực quan hóa phân cụm sau khi giảm chiều

5.1.1 Phương pháp Elbow

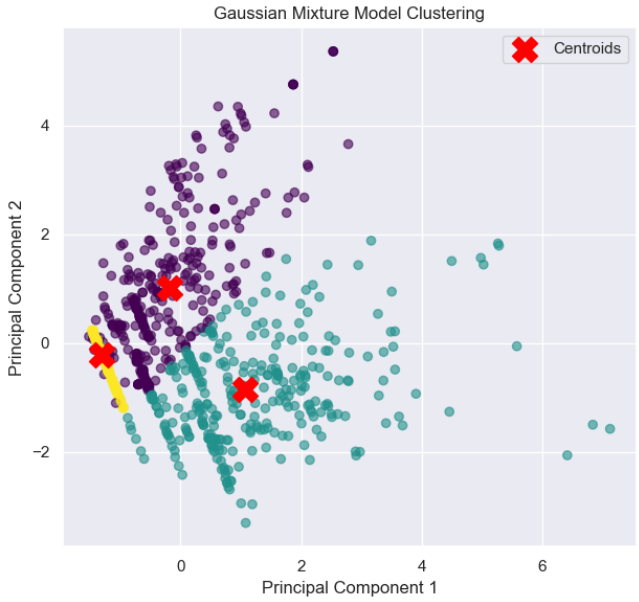


⇒ Nhận xét: Điểm khuỷ tay là điểm mà ở đó tốc độ suy giảm của hàm biến dạng sẽ thay đổi nhiều nhất. Tức là kể từ sau vị trí này thì gia tăng thêm số lượng cụm cũng không giúp hàm biến dạng giảm đáng kể. Trong hình trên thì ta có thể thấy tại điểm $k = 3$ là nơi xuất hiện điểm khuỷ tay rõ ràng nhất. Việc thêm cụm nữa sẽ không mang lại lợi ích đáng kể trong việc giảm tổng bình phương khoảng cách.

5.1.2 K-means

Hình 15: Phương pháp K-means

5.1.3 Gaussian mixture models



Hình 16: Phương pháp gmm

5.1.4 Kết luận

- K-Means tạo ra các cụm hình cầu, trong khi GMM có thể tạo ra các cụm có hình dạng elip. Điều này có thể dẫn đến sự khác biệt trong cách các cụm được xác định.
- GMM cho phép các cụm có kích thước khác nhau, trong khi K-Means giả định các cụm có kích thước tương tự. Điều này ảnh hưởng đến cách các điểm dữ liệu được gán vào các cụm.
- MM xử lý tốt hơn các cụm có phân bố chồng lấn, trong khi K-Means có thể gặp khó khăn với các cụm có phân bố phức tạp và không đều.

5.2 Kết quả phân loại với Random Forest

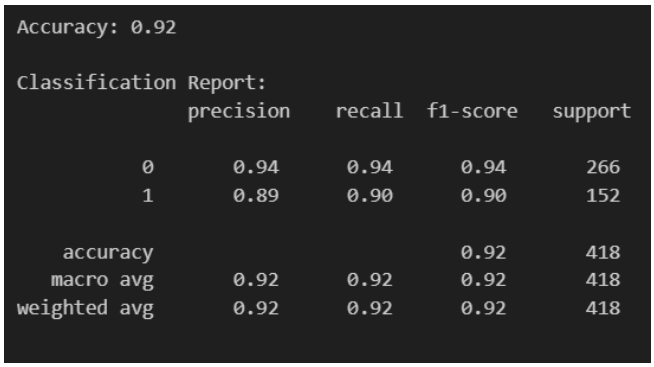
Nhóm có những cái thiện so với kết quả trước đây:

- Về tiền xử lý dữ liệu:
  - 'Age' được gán giá trị trung bình gom nhóm theo 'Pclass' và 'Sex' thay vì toàn bộ tập dữ liệu.
  - Bổ sung thêm trường thông tin 'Deck', phân hóa rõ hơn các hành khách có hạng 2 và 3.
- Về lựa chọn đặc trưng xây dựng mẫu bootstrap, bổ sung thêm 'Deck' và 'Age'.

Vậy nên, kết quả đánh giá Accuracy có sự cải thiện rõ rệt, tăng từ 81% lên 92%.

```
Accuracy:  0.8159371492704826
Precision:  0.8006756756756757
Recall:  0.6929824561403509
F1 Score:  0.74294670846395
```

Hình 17: Kết quả phân loại Random Forest trước tinh chỉnh



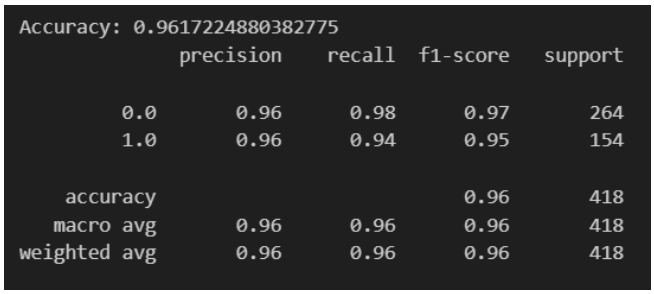
Hình 18: Kết quả phân loại Random Forest sau tinh chỉnh

5.3 Kết quả phân loại với kNN

Nhóm thực hiện đánh giá dựa trên các giá trị k khác nhau, bao gồm k bằng 1, 3, 5, 7, 9 và 11. Kết quả Accuracy quan sát ở bảng sau:

STT	Số lượng k	Accuracy
1	1	0.846
2	3	0.906
3	5	0.899
4	7	0.935
5	9	0.961
6	11	0.933

Có thể thấy với k = 9 thì cho kết quả Accuracy tốt nhất.



Hình 19: Kết quả phân loại kNN

Ngoài ra, nhóm thực hiện phương pháp xác thực chéo k-folds để đánh giá chính xác hơn hiệu suất mô hình. Với số lượng fold bằng 10, và đánh giá trên tập kiểm thử cho kết quả như sau:

STT	Số lượng k	Accuracy
1	1	0.9641
2	3	0.9593
3	5	0.9474
4	7	0.9403
5	9	0.9403
6	11	0.9474

Nguyên nhân có thể do:

- Khi sử dụng k-folds cross-validation, mỗi lần thử nghiệm mô hình sẽ chia dữ liệu thành các fold khác nhau. Khi k=1, mô hình sẽ chỉ xem xét một điểm gần nhất để dự đoán, điều này có thể phù hợp với cách mà tập dữ liệu được chia nhỏ thành các fold.

5.4 Kết luận

- Thay vì chỉ đơn giản xử lí giá trị bị thiếu bằng cách thông thường là tính trung bình các điểm dữ liệu trên toàn bộ tập dữ liệu thì việc gom nhóm và tính giá trị trung bình theo từng nhóm giúp cải thiện hơn mô hình về tổng thể.
- Xây dựng phương pháp PCA và áp dụng để giảm số chiều dữ liệu và hiển thị trực quan dữ liệu.
- Đánh giá và phân cụm với 2 phương pháp k-means và gaussian mixture model.
- Thực hiện 2 mô hình phân loại là Random Forest và k-Nearest Neighbors. Cả 2 mô hình đều cho kết quả khá tốt với Accuracy trên 90%. Tuy nhiên, với tập dữ liệu titanic thì kNN có vẻ vượt trội hơn.

Tài liệu tham khảo

[1] Rohan Bhargav, Pawan Whig. 2021. More Insight on Data Analysis of Titanic Data Set.

[2] Markus Ringnér. March 2008. What is principal component analysis?

[3] Aakash Parmar, Rakesh Katariya and Vatsal Patel. 21 December 2018. A Review on Random Forest: An Ensemble Classifier.

[4] Shichao Zhang, PictureXuelong Li, PictureMing Zong, PictureXiaofeng Zhu, PictureDebo Cheng. 12 January 2017. Learning k for kNN Classification.

[5] Rahul S. 2023. Machine Learning — Cluster Validation: The Elbow Method and Silhouette Score

[6] Aishwarya Singh. 2024. Build Better and Accurate Clusters with Gaussian Mixture Models.

[7] Adith Narasimhan Kumar. 2021. Why Linear Regression is not suitable for classification.

[8] Education Ecosystem (LEDU). 2018. Understanding K-means Clustering in Machine Learning.