Table of Contents

Chúng tôi, **Bùi Minh Huy**, **Trần Lê Vân**, **Nguyễn Thị Thanh Tâm** xin cam đoan rằng:

Tất cả thông tin và phân tích trình bày trong báo cáo này được thực hiện một cách chính xác và trung thực. Mọi dữ liệu, nhận định hoặc ý kiến được trích dẫn từ các nguồn khác đều đã được nêu rõ nguồn gốc và trích dẫn đúng quy định. Chúng tôi cam đoan rằng không có bất kỳ hành vi sao chép hoặc sử dụng thông tin không hợp pháp nào từ các nguồn khác. Bài báo cáo này là kết quả của công trình nghiên cứu độc lập của chúng tôi và chưa từng được công bố tại bất kỳ nơi nào khác. Chúng tôi cam đoan đã tuân thủ nghiêm ngặt các quy tắc và quy định của môn học, bao gồm việc tham khảo và áp dụng các công cụ nghiên cứu một cách hợp lệ. Nếu phát hiện có bất kỳ sự gian lận nào, chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung bài báo cáo của mình. Chúng tôi hy vọng rằng bài báo cáo này sẽ cung cấp những thông tin hữu ích cho các nhà nghiên cứu, doanh nghiệp, góp phần vào việc hiểu rõ hơn về mạng xã hội ngày nay.

Trong thời đại của điện toán di động và thiết bị thông minh, việc theo dõi và nhận dạng hoạt động con người (Human Activity Recognition - HAR) đã trở thành một lĩnh vực nghiên cứu đóng vai trò quan trọng trong nhiều ngành như trí tuệ nhân tạo, khoa học dữ liệu, y học và công nghệ cảm biến. HAR đóng vai trò cốt lõi trong các ứng dụng như giám sát sức khỏe, phát hiện té ngã, điều khiển nhà thông minh. Ngày nay, nhu cầu càng ngày gia tăng về các thiết bị công nghệ có thể hiểu hành vi con người dẫn đến việc phát triển các mô hình HAR là chính xác, hiệu quả và có khả năng triển khai thực tế là vô cùng cần thết.

Một trong những yếu tố chính thúc đẩy sự phát triển của HAR là sự phổ biến của các thiết bị di động thông minh và đồng hồ thông minh, vốn đợc trang bị sẵn các cảm biến quán tính bao gồm gia tốc kế (accelerometer) và con quay hồi chuyển (gyroscope). Những cảm biến này cho phép thu thập dữ liệu về chuyển động của người sử dụng với độ chính xác cao, chi phí thấp và tính khả dụng cao trong đời sống hàng ngày. Nhờ vậy, hệ thống HAR có thể được lắp đặt mà không cần sử dụng các thiết bị đắt tiền hoặc lắp đặt phức tạp.

Bên cạnh tiềm năng ứng dụng rộng rãi, việc xây dựng các mô hình HAR vẫn có nhiều khó khăn thách thức như dữ liệu cảm biến thường có số chiều lớn, có nhiều dữ liệu nhiễu và có tính biến động cao do phụ thuộc vào thói quen và hình thể của mỗi người. Bên cạnh đó, một số hoạt động có thể có mẫu tín hiệu tương tự nhau khiến cho các bài toán phân loại trở nên khó khăn hơn. Vì vậy, cần có một quy trình xử lý dữ liệu bài bản bao gồm các bước tiền xử lý dữ liệu, giảm chiều dữ liệu và huấn luyện mô hình học máy để có thể đặt được hiệu quả cao trong việc nhận dạng hoạt động con người.

Trong nghiên cứu này, chúng em đã tiến hành khai thác bộ dữ liệu “**Human Activity Recognition with Smartphones**” do UCI Machine Learning Repository cung cấp, một bộ dữ liệu được sử dụng rộng rãi trong cộng đồng nghiên cứu HAR. Chúng em đề xuất một quy trình học máy toàn diện bao gồm phân tích đặc trưng, giảm chiều dữ liệu bằng UMAP, PCA, TSNE và huấn luyện bằng các mô hình học máy như Random Forest, Decision Tree, Logistic Regression, Support Vectot Machine (SVM) để phân loại các hoạt động với mục tiêu là nâng cao độ chính xác và hiệu quả của mô hình. Những kết quả này sẽ cung cấp cái nhìn thực nghiệm rõ ràng cho các nhà nghiên cứu, đồng thời làm nền móng cho việc triển khai các hệ thống nhận dạng hoạt động trong thế giới thực.

## 0.1 Hồi quy logistic (Logistic Regression)

### 0.1.1 Khái niệm

Hồi quy logistic (Logistic Regression) là một thuật toán học máy có giám sát (supervised learning), đồng thời cũng là một phương pháp thống kê phổ biến, được sử dụng rộng rãi trong việc phân tích và dự đoán dữ liệu phân loại. Mô hình này đặc biệt hiệu quả trong các bài toán phân loại nhị phân, nơi mà biến phụ thuộc chỉ có hai giá trị khả dĩ như có/không, đúng/sai hoặc 1/0.[1]

Hồi quy logistic thường được ưu tiên sử dụng trong các bài toán dự đoán khi biến phụ thuộc không phải là biến liên tục mà là biến nhị phân hoặc phân loại, giúp cung cấp cái nhìn định lượng và chính xác về mối quan hệ giữa các biến độc lập và xác suất xảy ra của sự kiện cần phân tích.

### 0.1.2 Các loại hồi quy logistic

**Hồi quy logistic nhị phân (Binary Logistic Regression)**. Hồi quy logistic nhị phân dự đoán mối quan hệ giữa các biến phụ thuộc nhị phân và độc lập . Một số ví dụ về đầu ra của loại hồi quy này có thể là thành công/thất bại, 0/1 hoặc đúng/sai.

**Hồi quy logistic đa thức : (Multinomial Logistic Regression)**. Biến phụ thuộc phân loại có hai hoặc nhiều kết quả rời rạc trong loại hồi quy đa thức. Hồi quy logistic đa thức có nhiều hơn hai kết quả có thể xảy ra .

**Hồi quy logistic thứ tự (Ordinal Logistic Regression)**. Hồi quy logistic thứ tự áp dụng khi biến phụ thuộc ở trạng thái có thứ tự (tức là thứ tự.

### 0.1.3 Hàm Sigmoid

Hồi quy logistic dự đoán xác suất rơi vào một trong hai lớp (binary classification), thường được ký hiệu là 0 hoặc 1.

Để biểu diễn xác suất này, sử dụng hàm sigmoid, có dạng S-shaped và giới hạn giá trị đầu ra trong khoảng từ 0 đến 1.

Công thức của hàm sigmoid:

Trong đó:

* s(z) = đầu ra trong khoảng từ 0 đến 1 (giá trị xác suất ước lượng).
* z = đầu vào của hàm (giá trị dự đoán của thuật toán, ví dụ như mx+b).
* e = hằng số Euler, và là cơ sở của logarithm tự nhiên.

**Đặc điểm mô hình:**

* Phân lớp và dự đoán: Dự đoán biến phụ thuộc nhị phân hoặc danh mục từ một hoặc nhiều biến độc lập.
* Xác định mức độ ảnh hưởng của biến độc lập: Xác định cách thức và mức độ mà các biến độc lập ảnh hưởng đến xác suất của sự kiện hoặc lớp mục tiêu.
* Tính toán xác suất sự kiện: Cung cấp ước lượng xác suất cho một sự kiện xảy ra dựa trên biến độc lập.

### 0.1.4 Cách thức hoạt động của mô hình

Hồi quy logistic sử dụng hàm logistic (còn gọi là hàm sigmoid) để chuyển đổi giá trị dự đoán thành xác suất. Hàm logistic có dạng:

Công thức của mô hình hồi quy logistic:

Trong đó:

*P(Y = 1)* là xác suất để sự kiện *Y = 1* xảy ra (ví dụ: sự kiện thành công, top 1,…). *X₁, X₂, …, Xₖ* là các biến độc lập. *β₀, β₁, …, βₖ* là hệ số mô hình cần được ước lượng. *e* là cơ số của logarithm tự nhiên.

**Cách hoạt động**

* **Ước lượng hệ số mô hình**: Hệ số β của mô hình được ước lượng thông qua quy trình tối ưu hóa, thường là phương pháp Maximum Likelihood Estimation (MLE).MLE tìm cách tối đa hóa xác suất của dữ liệu quan sát dựa trên hệ số β.
* **Phân lớp**: Dựa vào xác suất được dự đoán từ hàm logistic, quyết định phân loại một quan sát vào lớp 1 nếu P(Y = 1) ≥ một ngưỡng cụ thể (thường là 0.5) và ngược lại là lớp 0. Ví dụ: Nếu P(Y = 1) > 0.5, quan sát được phân loại là lớp 1.
* **Đánh giá mô hình:** Mô hình hồi quy logistic thường được đánh giá thông qua các chỉ số như độ chính xác (accuracy), precision, recall, điểm số F1, hoặc thống qua ROC và AUC.

### 0.1.5 Nhược điểm của hồi quy logistic:

* Không thích hợp với biến phụ thuộc liên tục.
* khó khăn trong việc mô hình hóa mối quan hệ phức tạp hoặc không tuyến tính mà không cần biến đổi dữ liệu.
* Không hiệu quả khi xử lý dữ liệu có nhiều biến độc lập hoặc có sự tương quan cao giữa các biến.

## 0.2 K-Nearest Neighbors (KNN)

### 0.2.1 Khái niệm

K-Nearest Neighbors (KNN) là một trong những thuật toán học máy đơn giản nhất nhưng hiệu quả, thuộc nhóm học có giám sát (supervised learning). Thuật toán được sử dụng cho cả hai bài toán phân lớp (classification) và hồi quy (regression), tuy nhiên nó phổ biến hơn trong các bài toán phân lớp. Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới.[2]

### 0.2.2 Cách thức hoạt động của thuật toán KNN

Bước 1: Chọn một số nguyên K (số lượng hàng xóm gần nhất cần xét).

Bước 2: Tính khoảng cách của data input với các data trong có trong tập data train, có các cách tính khoảng cách như: Minkowski, Euclid, Manhattan, … tùy mục đích sử dụng mà chúng ta sử dụng cách tính khoảng cách, thông dụng nhất là cách tính Euclid.

Bước 3: Sau khi tính khoảng cách từ data input tới toàn bộ data trong tập training, chọn ra K lân cận với khoảng cách ngắn nhất, với K được chọn ở bước số 1.

Bước 4: Thực hiện phân loại, kết quả sẽ theo label có tỉ lệ voting cao nhất.

**Lựa chọn giá trị K**

* Giá trị K quá nhỏ (ví dụ K = 1): mô hình nhạy cảm với nhiễu (noise), dễ bị overfitting.
* Giá trị K quá lớn: mô hình quá “mềm”, dễ bị underfitting

### 0.2.3 Ưu điểm

* Đơn giản, dễ hiểu: Thuật toán có nguyên lý hoạt động trực quan, dễ cài đặt.
* Không cần huấn luyện: Không tốn thời gian xây dựng mô hình.
* Hiệu quả với dữ liệu nhỏ: Hoạt động tốt với tập dữ liệu kích thước vừa phải.
* Ứng dụng đa dạng: Có thể áp dụng cho cả bài toán phân lớp và hồi quy.
* Không có giả định về dữ liệu: Là mô hình phi tham số, không yêu cầu dữ liệu tuân theo phân phối cụ thể.
* Thích ứng tốt với dữ liệu mới: Dễ dàng cập nhật mô hình bằng cách thêm mẫu mới.

### 0.2.4 Nhược điểm

* Chi phí dự đoán cao: Phải tính khoảng cách đến mọi mẫu dữ liệu, tốn kém với dữ liệu lớn.
* Nhạy cảm với quy mô dữ liệu: Các đặc trưng có phạm vi lớn sẽ áp đảo các đặc trưng có phạm vi nhỏ.
* Nhạy cảm với nhiễu và dữ liệu thừa: Kém hiệu quả khi có nhiều đặc trưng không liên quan.
* Vấn đề với dữ liệu không cân bằng: Các lớp thịnh hành có xu hướng áp đảo các lớp thiểu số.
* Khó chọn k tối ưu: Giá trị k thích hợp phụ thuộc vào dữ liệu cụ thể.

## 0.3 Support Vector Machine (SVM)

### 0.3.1 Khái niệm

SVM là một thuật toán học máy có giám sát (supervised learning), dùng để giải quyết các bài toán phân loại và hồi quy.Mục tiêu chính của SVM là tìm một siêu phẳng (hyperplane) tốt nhất để phân tách các điểm dữ liệu thuộc hai lớp khác nhau sao cho khoảng cách (margin) giữa siêu phẳng và các điểm gần nhất của mỗi lớp là lớn nhất.Điểm dữ liệu gần nhất đó được gọi là vector hỗ trợ (support vector). SVM sử dụng các hàm kernel để biểu diễn không gian dữ liệu ban đầu vào không gian cao chiều hơn,giúp phân loại tốt hơn đối với các bài toán phức tạp.[3]

### 0.3.2 Các loại SVM

Trong thuật toán SVM, có hai loại chính là Linear SVM (SVM tuyến tính) và Non-linear SVM (SVM phi tuyến).

Linear SVM là loại SVM mà ta có thể phân chia dữ liệu bằng một đường thẳng. Điều này áp dụng cho các bài toán có dữ liệu tuyến tính và có thể tách biệt bằng một đường thẳng.

Non-linear SVM được sử dụng khi không thể phân chia dữ liệu bằng một đường thẳng. Trong trường hợp này, ta sẽ sử dụng các phương pháp biến đổi dữ liệu sao cho chúng trở thành tuyến tính, sau đó áp dụng Linear SVM để giải quyết bài toán.

### 0.3.3 Cách hoạt động

* Chọn siêu phẳng phân tách: SVM tìm kiếm siêu phẳng sao cho có khoảng cách margin lớn nhất giữa hai lớp dữ liệu.
* Tối đa hóa margin: Cố gắng duy trì khoảng cách lớn nhất từ siêu phẳng tới các điểm dữ liệu gần nhất.
* Áp dụng kernel nếu cần: Nếu dữ liệu không phân tách tuyến tính, áp dụng kernel trick để biến đổi không gian dữ liệu.
* Đánh giá và điều chỉnh tham số để tối ưu mô hình.

### 0.3.4 Margin trong SVM

Margin là khoảng cách từ siêu phẳng phân tách (hyperplane) đến các điểm dữ liệu gần nhất thuộc hai lớp khác nhau – các điểm này được gọi là vector hỗ trợ (support vectors). Trong hình dung đơn giản, ví dụ như bài toán phân loại quả táo và quả lê đặt trên mặt bàn, margin chính là khoảng cách từ cây que (đại diện cho siêu phẳng) đến quả táo và quả lê gần cây que nhất. Điều quan trọng trong SVM là thuật toán luôn tìm cách tối đa hóa margin này, nhằm tạo ra một siêu phẳng phân tách có khoảng cách lớn nhất đến các điểm dữ liệu gần ranh giới nhất của mỗi lớp. Nhờ đó, SVM có thể tăng cường khả năng tổng quát hóa và giảm thiểu nguy cơ phân loại sai khi xử lý các điểm dữ liệu mới chưa từng thấy trước đó.

### 0.3.5 Thủ thuật Kernel

Kernel là một hàm ánh xạ dữ liệu từ không gian ít nhiều hơn sang không gian nhiều chiều hơn, từ đó ta tìm được siêu phẳng phân tách dữ liệu. Một cách trực quan, kỹ thuật này giống như việc bạn gập tờ giấy lại để có thể dùng kéo cắt một lỗ tròn trên nó.

### 0.3.6 Ưu điểm

* Xử lý trên không gian số chiều cao: SVM là một công cụ tính toán hiệu quả trong không gian chiều cao, trong đó đặc biệt áp dụng cho các bài toán phân loại văn bản và phân tích quan điểm nơi chiều có thể cực kỳ lớn.
* Tiết kiệm bộ nhớ: Do chỉ có một tập hợp con của các điểm được sử dụng trong quá trình huấn luyện và ra quyết định thực tế cho các điểm dữ liệu mới.
* Tính linh hoạt - phân lớp thường là phi tuyến tính. Khả năng áp dụng Kernel mới phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiếncho hiệu suất phân loại lớn hơn.

### 0.3.7 Nhược điểm

* Thuật toán SVM có độ phức tạp tính toán cao khi số lượng dữ liệu lớn.
* SVM yêu cầu dữ liệu huấn luyện là tuyến tính hoặc phi tuyến tính.
* Thuật toán SVM cần lựa chọn tham số tốt để đạt được kết quả tốt nhất.

## 0.4 Decision Tree

### 0.4.1 Khái niệm

Cây quyết định (Decision Tree) là một trong những thuật toán phổ biến nhất trong lĩnh vực máy học (Machine Learning). Là một công cụ mạnh mẽ, thường được sử dụng để giải quyết các bài toán về phân loại (classification) và dự đoán (regression) trong khai phá dữ liệu.Cây quyết định là một thuật toán máy học dùng để dự đoán hoặc phân loại dữ liệu dựa trên các bước ra quyết định nối tiếp nhau.[4]

### 0.4.2 Cách hoạt động

Cách hoạt động của cây quyết định rất đơn giản:

* Đầu tiên, cây quyết định sẽ xem xét toàn bộ tập dữ liệu.
* Sau đó, chia dữ liệu thành các nhóm nhỏ dựa trên một số đặc điểm hoặc điều kiện nhất định (ví dụ như tuổi tác, thu nhập, màu sắc, nhiệt độ..).
* Việc chia nhỏ này sẽ tiếp tục thực hiện đến khi dữ liệu ở mỗi nhóm trở nên rõ ràng và dễ dàng để dự đoán hoặc phân loại.
* Trong cây quyết định, mỗi “nút” (node) đại diện cho một đặc điểm (ví dụ: “tuổi”, “mức thu nhập”, “thời tiết”). Các “nhánh” (branch) nối giữa các nút chính là các điều kiện để chia dữ liệu (ví dụ: tuổi lớn hơn hay nhỏ hơn 30). Cuối cùng, các “nút lá” (leaf node) là kết quả cuối cùng mà cây quyết định đưa ra, ví dụ như “mua hàng” hay “không mua hàng”, “đi chơi thể thao” hay “ở nhà”.
* Nói cách khác, cây quyết định giống như một chuỗi các câu hỏi đơn giản được sắp xếp theo từng bước, giúp ta dễ dàng đi đến một quyết định cuối cùng.

### 0.4.3 Công thức

**Gini Impurity**

Công thức Gini Impurity được sử dụng trong thuật toán cây quyết định để đo lường độ không chính xác của một dự đoán khi phân loại một tập dữ liệu.

Cần lưu ý là Gini Impurity càng nhỏ (gần 0) thì tập dữ liệu đó càng “thuần khiết”, nghĩa là các mẫu trong cùng một nhóm có xu hướng thuộc vào cùng một lớp. Ngược lại, nếu Gini Impurity cao (gần 1), thì việc phân loại các mẫu trong nhóm đó trở nên không chắc chắn.

Giả sử khi đang xem xét một tập dữ liệu chia thành K nhóm, mỗi nhóm chứa một phần tỷ lệ pi với i=1,2,…,K.

Công thức tính độ bất thuần Gini (Gini Impurity):

Trong đó:

* là Gini Impurity.
* là tỷ lệ các mẫu thuộc vào lớp .
* Khi xây dựng cây quyết định, chúng ta cần chọn thuộc tính và giá trị phân chia sao cho Gini Impurity sau phân chia là nhỏ nhất, tức là mức độ “thuần khiết” cho dữ liệu thuộc về từng nhóm con được tạo ra.

**Entropy**

Entropy trong cây quyết định là một khái niệm được sử dụng để đo lường sự không chắc chắn trong dữ liệu (Trung bình surprise). Trong ngữ cảnh của cây quyết định, entropy thường được sử dụng để đo lường mức độ không chắc chắn của phân phối lớp trong tập dữ liệu.

Entropy được tính bằng công thức sau:

Trong đó:

* là tập dữ liệu.
* là số lớp trong tập dữ liệu.
* là tỷ lệ của lớp trong tập dữ liệu.
* Entropy càng cao khi tỷ lệ của các lớp trong tập dữ liệu gần bằng nhau, và càng thấp khi một lớp chiếm đa số.
* Khi xây dựng cây quyết định, chúng ta cố gắng chia tập dữ liệu sao cho entropy sau khi chia là thấp nhất có thể. Điều này giúp cây quyết định có thể học được các quy tắc quyết định hiệu quả từ dữ liệu.
* Quyết định về cách chia tập dữ liệu dựa trên entropy thường được thực hiện bằng cách so sánh entropy trước và sau khi chia, và chọn cách chia mà giảm entropy nhiều nhất.

### 0.4.4 Ưu điểm

* Mô hình sinh ra các quy tắc dễ hiểu cho người đọc, tạo ra bộ luật với mỗi nhánh lá là một luật của cây.
* Dữ liệu đầu vào có thể là dữ liệu missing, không cần chuẩn hóa hoặc tạo biến giả.
* Có thể làm việc với cả dữ liệu số và dữ liệu phân loại.
* Có thể xác thực mô hình bằng cách sử dụng các kiểm tra thống kê.
* Có khả năng làm với dữ liệu lớn.

### 0.4.5 Nhược điểm

* Mô hình cây quyết định phụ thuộc rất lớn vào dữ liệu. Thậm chí, 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 hay gặp vấn đề overfitting

### 0.4.6 Các vấn đề sau khi thực hiện áp dụng mô hình vào dữ liệu cần dự đoán

**Underfitting:**là hiện tượng kết quả độ chênh lệch của mô hình được huấn luyện và kết quả độ chênh lệch của dữ liệu cần dự đoán đạt giá trị mức cao giống nhau, do mô hình chưa được huấn luyện đầy đủ. Cần xem lại cấu trúc của mô hình (tăng thêm độ phức tạp) để có thể huấn luyện các tập dữ liệu khó và tăng thêm dữ liệu huấn luyện để tăng hiệu suất của mô hình.

**Overfitting:**là hiện tượng kết quả của mô hình được huấn luyện quá tốt (độ chênh lệch thấp) nhưng khi áp dụng vào dữ liệu cần dự đoán thì mô hình đạt hiêu suất kém (độ chênh lệch cao) do mô hình đã học quát sát với dữ liệu huấn luyện và không có khả năng tổng quát hóa các dữ liệu cần dự đoán. Cần sử dụng một số các phương pháp tránh overfitting như tăng độ đa dạng của dữ liệu, giảm thiểu độ phức tạp của mô hình.

=> Giải pháp : Pruning solution là được áp dụng đối với trường hợp mô hình huấn uyện bị overfitting khi sử dụng mô hình Decision Tree bằng cách hạn chế kích thước, chiều sâu của mô hình này.

## 0.5 Random Forest

### 0.5.1 Khái niệm

Random Forest là một thuật toán học máy thuộc nhóm học có giám sát (supervised learning), được sử dụng phổ biến trong cả bài toán phân loại và hồi quy.Khác với Decision Tree chỉ dựa vào một cây duy nhất, Random Forest kết hợp nhiều cây quyết định để tạo ra một mô hình tổng hợp, có khả năng dự đoán chính xác và ổn định hơn.[5]

### 0.5.2 Cách hoạt động của Random Forest

* Tạo ra nhiều cây khác nhau từ các tập dữ liệu nhỏ, được chọn ngẫu nhiên từ tập dữ liệu gốc.
* Mỗi cây sẽ đưa ra dự đoán riêng.
* Mô hình sẽ lấy trung bình (với hồi quy) hoặc bỏ phiếu theo số đông (với phân loại) để đưa ra kết quả cuối cùng.

### 0.5.3 Công thức

Bài toán phân loại (classification) Bài toán phân loại, mỗi cây quyết định trong rừng sẽ đưa ra một nhãn dự đoán. Sau đó, mô hình Random Forest sẽ lấy nhãn xuất hiện nhiều nhất trong các dự đoán của từng cây làm kết quả cuối cùng.

Công thức được biểu diễn như sau:

Trong đó:

* là kết quả dự đoán cuối cùng của mô hình.
* là kết quả dự đoán của cây thứ với đầu vào .
* là tổng số cây trong mô hình Random Forest.
* *mode* là hàm chọn giá trị xuất hiện nhiều nhất.
* Bài toán hồi quy (Regression Với bài toán hồi quy, mỗi cây trong rừng sẽ đưa ra một giá trị số. Kết quả cuối cùng được tính bằng cách lấy trung bình cộng các giá trị dự đoán của tất cả các cây.

Được tính bằng công thức sau:

Trong đó:

* là giá trị dự đoán cuối cùng.
* là giá trị dự đoán từ cây thứ .
* là tổng số cây trong mô hình.

### 0.5.4 Ưu điểm

* Độ chính xác cao: mô hình tổng hợp nhiều cây, Random Forest thường cho kết quả chính xác hơn so với cây quyết định đơn lẻ.
* Chống overfitting tốt: kết hợp nhiều cây được huấn luyện từ dữ liệu và thuộc tính ngẫu nhiên giúp mô hình tránh học quá sát vào dữ liệu huấn luyện.
* Làm việc tốt với dữ liệu lớn và có nhiều đặc trưng: Random Forest xử lý hiệu quả dữ liệu có số chiều lớn và phức tạp.
* Không yêu cầu chuẩn hóa dữ liệu: Dữ liệu đầu vào không cần phải được chuẩn hóa hoặc xử lý đặc biệt như một số thuật toán khác.
* Có thể đo lường mức độ quan trọng của các thuộc tính (feature importance): Giúp phân tích và chọn ra các yếu tố ảnh hưởng nhiều nhất đến kết quả dự đoán.

### 0.5.5 Nhược điểm

* Khó giải thích mô hình: Random Forest gồm nhiều cây nên khó để xem toàn bộ quátrình mô hình đưa ra kết quả , khó để giải thích được mô hình .
* Thời gian huấn luyện và dự đoán lâu hơn: Mô hình gồm nhiều cây nên tốn nhiều thời gian và tài nguyên hơn khi xử lý dữ liệu lớn.
* Chiếm nhiều bộ nhớ: Lưu trữ nhiều cây có thể tốn nhiều RAM, đặc biệt khi số lượng cây lớn.
* Dễ bị bias nếu dữ liệu không cân bằng: Trong trường hợp dữ liệu bị lệch, Random Forest có thể dự đoán thiên lệch theo lớp đó nếu không xử lý cân bằng dữ liệu trước.

## 0.6 UMAP

### 0.6.1 Khái niệm

**Uniform Manifold Approximation and Projection (UMAP)** là một kỹ thuật giảm chiều dữ liệu, tương tự như t-SNE, thường được sử dụng để trực quan hóa dữ liệu.[6] Ngoài ra, UMAP còn có thể được sử dụng như một phương pháp giảm chiều phi tuyến tổng quát trong các bài toán học máy. Thuật toán UMAP được xây dựng dựa trên ba giả định chính về cấu trúc dữ liệu: 1. Dữ liệu phân bố đều trên một đa tạp Riemannian (Riemannian manifold); 2. Metric Riemannian là hằng số cục bộ (hoặc có thể được xấp xỉ là như vậy); 3. Ống phân phối (local connectivity) được kết nối cục bộ.

### 0.6.2 Mục tiêu của UMAP

• Giảm số lượng biến: UMAP chuyển đổi dữ liệu sang một không gian có số chiều thấp hơn, phù hợp cho các tác vụ xử lý và phân tích tiếp theo.

* Giữ lại cấu trúc dữ liệu: UMAP cố gắng bảo toàn cả cấu trúc cục bộ (local structure) lẫn cấu trúc tổng thể (global structure) của dữ liệu trong không gian mới.
* Bảo tồn mối quan hệ phi tuyến: Khác với PCA, UMAP có khả năng nắm bắt và biểu diễn các mối quan hệ phi tuyến giữa các điểm dữ liệu.
* Trực quan hóa dữ liệu: UMAP đặc biệt hiệu quả trong việc trực quan hóa dữ liệu nhiều chiều trong không gian 2D hoặc 3D, với độ chính xác và sắc nét cao hơn so với nhiều kỹ thuật khác như t-SNE.

### 0.6.3 Quy trình thực hiện UMAP

* Chuẩn hóa dữ liệu (nếu cần).
* Tìm k hàng xóm gần nhất cho mỗi điểm và xây dựng đồ thị fuzzy biểu diễn cấu trúc cục bộ.
* Tính toán xác suất kết nối giữa các điểm dựa trên khoảng cách và hàm kernel.
* Khởi tạo các điểm trong không gian thấp chiều và tối ưu hóa đồ thị sao cho bảo toàn cấu trúc so với đồ thị ban đầu.
* Chiếu dữ liệu sang không gian mới để phục vụ trực quan hóa hoặc các bước phân tích tiếp theo.

### 0.6.4 Ưu điểm

* Bảo toàn cấu trúc cục bộ tốt.
* Có thể dùng cho cả supervised & unsupervised.
* UMAP cho phép lưu và áp dụng mô hình lên dữ liệu mới.

### 0.6.5 Nhược điểm

* Không giải thích được biến gốc.
* Phụ thuộc vào tham số.

Bộ dữ liệu **Human Activity Recognition with Smartphones** được xây dựng nhằm phục vụ cho các nghiên cứu về nhận diện hành vi con người thông qua dữ liệu cảm biến thu thập từ thiết bị di động. Tập dữ liệu được thu thập từ 30 người tham gia (gọi là *subjects*) (15 nam và 15 nữ, độ tuổi từ 19 đến 48) thực hiện sáu hoạt động thường ngày như Đi bộ (Walking), đi lên cầu thang (Walking Upstairs), đi xuống cầu thang (WalkingDownstairs), ngồi (Sitting), đứng (Standing)và nằm (Laying).[7]

Dữ liệu được ghi lại bằng một điện thoại thông minh (Samsung Galaxy S II) đeo ở thắt lưng của người dùng. Thiết bị này sử dụng hai loại cảm biến là **accelerometer** và **gyroscope** được sử dụng để ghi lại chuyển động theo 3 trục X, Y, Z với tần số lấy mẫu 50Hz.

Mỗi chuỗi tín hiệu được chia thành các **cửa sổ trượt** có độ dài 2.56 giây, tương ứng với 128 lần đo. Từ mỗi cửa sổ, các đặc trưng (features) đã được trích xuất từ **miền thời gian** và **miền tần số** để tạo ra một tập hợp dữ liệu có cấu trúc sẵn sàng cho mô hình học máy.

Các tín hiệu chính được sử dụng để tạo đặc trưng bao gồm:

* tBodyAcc-XYZ: Gia tốc cơ thể theo 3 trục trong miền thời gian
* tGravityAcc-XYZ: Gia tốc do trọng lực
* tBodyGyro-XYZ: Tốc độ quay từ con quay hồi chuyển
* tBodyAccJerk-XYZ, tBodyGyroJerk-XYZ: Jerk - đo sự thay đổi đột ngột của chuyển động
* Mag: Độ lớn vector gia tốc, được tính bằng chuẩn Euclidean:
* fBodyAcc-XYZ, fBodyGyro-XYZ: Biến miền tần số được tạo từ FFT

Từ các tín hiệu trên, 561 đặc trưng thống kê đã được trích xuất, bao gồm:

* mean(), std(), mad(), max(), min(): Đặc trưng thông kê truyền thống
* sma(): Signal Magnitude Area
* energy(): Tổng bình phương chia số phần tử
* entropy(), iqr(), arCoeff(), correlation()
* meanFreq(), skewness(), kurtosis(), bandsEnergy(), angle()
* Toàn bộ bộ dữ liệu được chia thành hai phần: Tập huấn luyện (train.csv): bao gồm 7352 mẫu. Tập kiểm tra (test.csv): bao gồm 2947 mẫu. Mỗi mẫu tương ứng với một cửa sổ thời gian 2.56 giây, được biểu diễn bằng **561 đặc trưng đầu vào (features)**, cùng với một mã định danh người thực hiện (subject) và một nhãn hoạt động (Activity), các nhãn này được mã hóa từ 1 đến 6 tương ứng với:

| Giá trị nhãn | Hoạt động |
| --- | --- |
| 1 | WALKING |
| 2 | WALKING\_UPSTAIRS |
| 3 | WALKING\_DOWNSTAIRS |
| 4 | SITTING |
| 5 | STANDING |
| 6 | LAYING |

# Tài liệu tham khảo

[1] “FrontMatter,” in *Applied Logistic Regression*, John Wiley & Sons, Ltd, 2013, pp. i–xvi. doi: [10.1002/9781118548387.fmatter](https://doi.org/10.1002/9781118548387.fmatter).

[2] “Nearest neighbor pattern classification IEEE Journals & Magazine IEEE Xplore.” Accessed: Apr. 15, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/1053964>

[3] “1.4. Support Vector Machines,” *scikit-learn*. Accessed: Apr. 15, 2025. [Online]. Available: <https://scikit-learn/stable/modules/svm.html>

[4] “A Complete Guide to Decision Trees Paperspace Blog,” *Paperspace by DigitalOcean Blog*. Feb. 2020. Accessed: Apr. 15, 2025. [Online]. Available: <https://blog.paperspace.com/decision-trees/>

[5] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324).

[6] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.” arXiv, Sep. 2020. doi: [10.48550/arXiv.1802.03426](https://doi.org/10.48550/arXiv.1802.03426).

[7] “Human Activity Recognition with Smartphones.” Accessed: Apr. 15, 2025. [Online]. Available: <https://www.kaggle.com/datasets/uciml/human-activity-recognition-with-smartphones>