**BÀI THỰC HÀNH 5: XÂY DỰNG MÔ HÌNH MÁY HỌC TRÊN DỮ LIỆU ĐÃ THU THẬP**

Bộ dữ liệu: **Human Activity Recognition Using Smartphones Data Set**

Link dữ liệu: [https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI HAR Dataset.zip](https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip)

**Ghi chú: Bộ dữ liệu này đã được tiền xử lý ở BTTH số 5 trước đó.**

Các thư viện cần thiết: caret, kernlab.

**Bài 1: Sử dụng lại bộ dữ liệu Human Activity Recognition Using Smartphones Data Set**

1) Đọc dữ liệu đã được tiền xử lý trước đó. Bỏ đi thuộc tính *Subject*.

2) Thống kê số lượng nhãn, số lượng thuộc tính và phân bố dữ liệu theo nhãn.

3) Chuẩn bị dữ liệu: phân chia dữ liệu ban đầu thành 2 tập: huấn luyện (train) và kiểm thử (test) với tỉ lệ 8-2.

4) Cài đặt mô hình: **kNN và SVM**. Sử dụng kỹ thuật cross validation.

5) Đánh giá mô hình: Độ chính xác (Accuracy) Ma trận nhầm lẫn (Confusion matrix).

6) Có nhận xét gì về hiệu quả giữa 2 mô hình?

7) Cài đặt thêm 2 thuật toán: Linear Discriminant Analysis (LDA) và Random Forest (RF). So sánh kết quả giữa 4 mô hình với nhau.

*8)\* Thực hiện lại các bước từ 1 tới 7 khoảng 5 lần, ứng với mỗi lần thực hiện ghi lại giá trị Accuracy và thay đổi giá trị seed. Sau đó tính và ghi ra kết quả Accuracy trung bình của 5 lần thực hiện.*

**Bài 2:** Thực hiện các yêu cầu trên với bộ dữ liệu **Iris**. Link: https://archive.ics.uci.edu/ml/datasets/iris

**Bài 3:** Thực hiện các yêu cầu trên với bộ dữ liệu **WineQuality**. Link: https://archive.ics.uci.edu/ml/datasets/Wine+Quality

**Nộp bài:**

**- File PDF báo cáo kết quả thực hiện đối với các mô hình máy học**

**- Source code thực hiện.**

Nén lại và đặt tên theo cú pháp **<MSSV>\_<Họ tên>\_BT6.rar**. Nộp qua course (Giảng viên sẽ tạo submission sau).

*Chúc tất cả các bạn học tốt*

# **Hướng dẫn thực hiện:**

**Bước 1: Khai báo 2 thư viện:**

library(caret)

library(kernlab)

**Bước 2: Đọc dữ liệu: sử dụng hàm read.csv()**

data <- read.csv("data/tidy\_data.csv")

**Bước 3: Thống kê dữ liệu**

- Số nhãn (labels): Sử dụng hàm **unique()** trên thuộc tính nhãn.

label <- unique(data$Activity)

- Thông tin tổng quan về các thuộc tính: sử dụng hàm **summary()**

summary(data)

- Phân bố dữ liệu trên từng nhãn: sử dụng **prop.table()**

percentage <- prop.table(table(data$Activity)) \* 100

cbind(freq=table(data$Activity), percentage=percentage)

**Bước 4: Chuẩn bị dữ liệu:**

- Phân chia train/test: sử dụng hàm **createDataPartition()**

validation\_index <- createDataPartition(data$Activity, p=0.80, list=FALSE)

**validation** <- data[-validation\_index,]

**training** <- data[validation\_index,]

- Thiết lập cross validation: sử dụng hàm trainControl(). Sử dụng thêm tham số savePredictions = TRUE để hiển thị kết quả huấn luyện theo từng fold.

control <- trainControl(method="**cv**", number=10, savePredictions = TRUE)

- Độ đo đánh giá: Accuracy

metric <- "Accuracy" # Do do danh gia

**Bước 5: Xây dựng và huấn luyện mô hình: sử dụng hàm train(), với method là thuật toán huấn luyện, metric là độ đo đánh giá.**

# kNN

set.seed(7)

fit.knn <- train(**Activity**~., data=**training**, method="**knn**", metric=**metric**, trControl=**control**)

# SVM

set.seed(7)

fit.svm <- train(**Activity**~., data=training, method="**svmRadial**", metric=**metric**, trControl=**control**)

**Bước 6: Phân tích kết quả:**

- Sử dụng phương pháp lấy mẫu: hàm resamples() để xem kết quả của mô hình trên dữ liệu huấn luyện.

results <- resamples(list(knn=**fit.knn**, svm=**fit.svm**))

summary(results)

- Xem kết quả dự đoán trên từng fold đối với mô hình KNN:

fit.knn$pred

- Dự đoán trên tập test: hàm predict()

predictions <- predict(fit.knn, validation)

- Phân tích kết quả dự đoán: sử dụng hàm: confusionMatrix()

confusionMatrix(predictions, as.factor(validation$Activity))

- Các tiêu chí nhận xét so sánh:

*1) Độ chính xác gữa các mô hình.*

*2) Hiệu quả dự đoán trên từng nhãn (dùng ma trận nhầm lẫn).*

*3) Nhãn nào dự đoán tốt nhất, nhãn nào dự đoán không tốt. Vì sao ?*

*4) Tỉ lệ nhầm lẫn của nhãn nào cao nhất.*

(Tham khảo thêm về cách đánh giá dựa trên confusion matrix tại đây: https://www.kdnuggets.com/2021/02/evaluating-deep-learning-models-confusion-matrix-accuracy-precision-recall.html)