38-UMAP

Bùi Minh Huy

Contents

1 Giới thiệu về UMAP

1.1 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. 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.
```

1.2 Mục tiêu của UMAP

UMAP có các mục tiêu sau:

- **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.

1.3 Quy trình thực hiện UMAP

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

2 Ứng dụng UMAP với bộ mtcars

2.1 Tải thư viện cần thiết

```
library(tidyverse)   # Bộ thư viện chứa nhiều công cụ xử lý dữ liệu như dplyr, tidyr, ggplot2,... để th
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)     # Thư viện tạo đồ thị với cú pháp ngữ pháp đồ họa (grammar of graphics) giúp tạo b
#install.packages(GGally)
library(GGally)      # Mở rộng của ggplot2, cung cấp các hàm để tạo ma trận tương quan, biểu đồ cặp (pa
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(factoextra)  # Thư viện cho phân tích đa chiều, hỗ trợ phân tích thành phần chính (PCA) và phân
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

2.2 Hiểu về bộ dữ liệu mtcars

```
data(mtcars)
# Xem cấu trúc bộ dữ liệu mtcars
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
# Hiển thị một số dòng đầu tiên
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
```

```
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
# Tóm tắt thống kê
summary(mtcars)
```

```
##       mpg             cyl             disp             hp
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
##       drat             wt             qsec             vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

Bộ dữ liệu mtcars chứa thông tin về 32 mẫu xe với 11 biến mô tả các đặc điểm kỹ thuật:

- mpg: Miles per gallon (hiệu suất tiêu thụ nhiên liệu)
- cyl: Số xi-lanh
- disp: Dung tích xi-lanh
- hp: Mã lực
- drat: Tỷ số truyền động sau
- wt: Trọng lượng (1000 lbs)
- qsec: Thời gian chạy 1/4 dặm
- vs: Kiểu động cơ (0 = chữ V, 1 = thẳng hàng)
- am: Kiểu hộp số (0 = tự động, 1 = số sàn)
- gear: Số lượng số
- carb: Số lượng bộ chế hòa khí

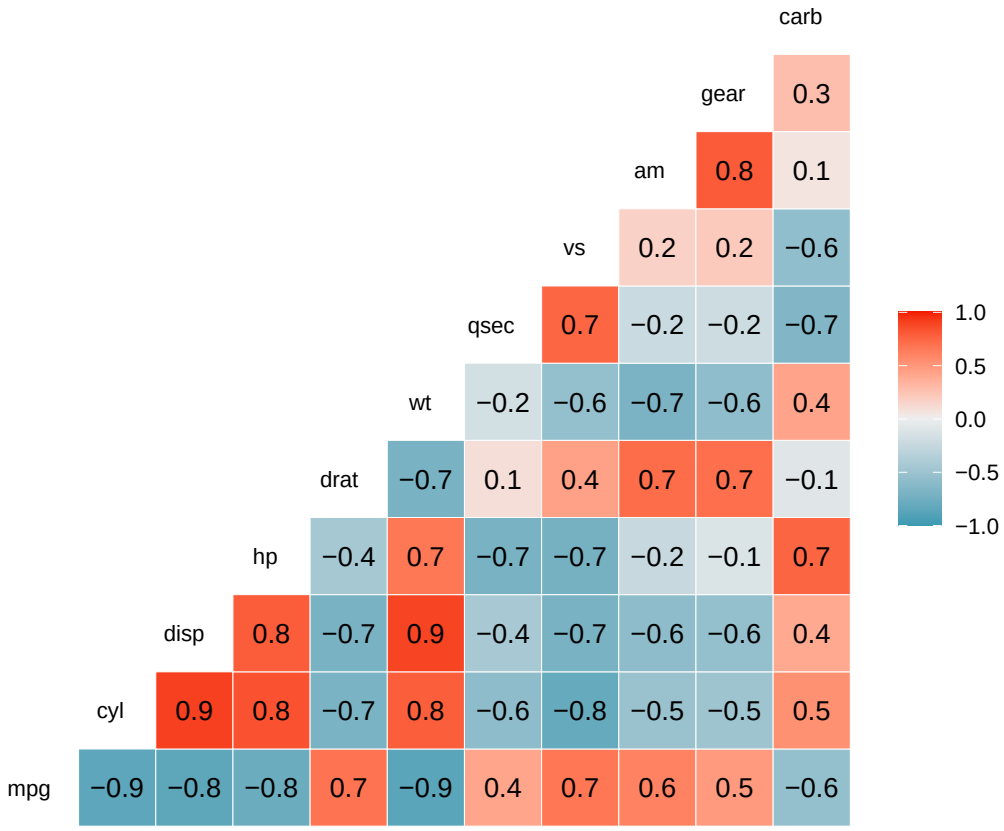
2.3 Phân tích tương quan

Trước khi thực hiện t-SNE, chúng ta hãy xem các biến có tương quan như thế nào:

```
# Tạo ma trận tương quan
cor_matrix <- cor(mtcars)
round(cor_matrix, 2)
```

```
##        mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
## mpg   1.00 -0.85 -0.85 -0.78  0.68 -0.87  0.42  0.66  0.60  0.48 -0.55
## cyl  -0.85  1.00  0.90  0.83 -0.70  0.78 -0.59 -0.81 -0.52 -0.49  0.53
## disp -0.85  0.90  1.00  0.79 -0.71  0.89 -0.43 -0.71 -0.59 -0.56  0.39
## hp   -0.78  0.83  0.79  1.00 -0.45  0.66 -0.71 -0.72 -0.24 -0.13  0.75
## drat  0.68 -0.70 -0.71 -0.45  1.00 -0.71  0.09  0.44  0.71  0.70 -0.09
## wt   -0.87  0.78  0.89  0.66 -0.71  1.00 -0.17 -0.55 -0.69 -0.58  0.43
## qsec  0.42 -0.59 -0.43 -0.71  0.09 -0.17  1.00  0.74 -0.23 -0.21 -0.66
## vs    0.66 -0.81 -0.71 -0.72  0.44 -0.55  0.74  1.00  0.17  0.21 -0.57
## am    0.60 -0.52 -0.59 -0.24  0.71 -0.69 -0.23  0.17  1.00  0.79  0.06
## gear  0.48 -0.49 -0.56 -0.13  0.70 -0.58 -0.21  0.21  0.79  1.00  0.27
## carb -0.55  0.53  0.39  0.75 -0.09  0.43 -0.66 -0.57  0.06  0.27  1.00
```

```
# Visualize correlation matrix
ggcorr(mtcars,
       method = c("everything", "pearson"),
       label = TRUE,
       hjust = 0.75,
       size = 3,
       layout.exp = 2)
```



Qua ma trận tương quan, chúng ta thấy nhiều biến có tương quan mạnh với nhau. Ví dụ: - cyl, disp, hp và wt có tương quan dương mạnh với nhau - Các biến trên có tương quan âm mạnh với mpg

Điều này chỉ ra rằng dữ liệu có thể có đa cộng tuyến và phù hợp để áp dụng UMAP.

2.4 Tiền xử lý dữ liệu

```
# Gán nhãn cho dữ liệu
mtcars_labeled <- mtcars %>%
  mutate(
    engine_type = factor(vs, labels = c("V-shaped", "Straight")),
    transmission = factor(am, labels = c("Automatic", "Manual")),
    cylinders = factor(cyl)
  )

# Chuẩn bị dữ liệu số cho UMAP
mtcars_features <- mtcars %>% select(-vs, -am)

# Chuẩn hóa dữ liệu
mtcars_scaled <- scale(mtcars_features)
head(mtcars_scaled)
```

```
##                           mpg        cyl        disp         hp       drat
## Mazda RX4           0.1508848 -0.1049878 -0.57061982 -0.5350928  0.5675137
```

```
## Mazda RX4 Wag      0.1508848 -0.1049878 -0.57061982 -0.5350928  0.5675137
## Datsun 710         0.4495434 -1.2248578 -0.99018209 -0.7830405  0.4739996
## Hornet 4 Drive     0.2172534 -0.1049878  0.22009369 -0.5350928 -0.9661175
## Hornet Sportabout -0.2307345  1.0148821  1.04308123  0.4129422 -0.8351978
## Valiant           -0.3302874 -0.1049878 -0.04616698 -0.6080186 -1.5646078
##                           wt       qsec       gear       carb
## Mazda RX4         -0.610399567 -0.7771651  0.4235542  0.7352031
## Mazda RX4 Wag     -0.349785269 -0.4637808  0.4235542  0.7352031
## Datsun 710        -0.917004624  0.4260068  0.4235542 -1.1221521
## Hornet 4 Drive    -0.002299538  0.8904872 -0.9318192 -1.1221521
## Hornet Sportabout  0.227654255 -0.4637808 -0.9318192 -0.5030337
## Valiant            0.248094592  1.3269868 -0.9318192 -1.1221521
```

2.5 Thực hiện UMAP

```
if (!require("uwot")) install.packages("uwot")
```

```
## Loading required package: uwot
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
library(uwot)

if (!require("ggplot2")) install.packages("ggplot2")
library(ggplot2)

engine_type <- mtcars_labeled$engine_type
transmission <- mtcars_labeled$transmission
cylinders <- mtcars_labeled$cylinders
car_names <- rownames(mtcars)

set.seed(42)
mtcar_umap <- umap(mtcars_scaled, n_neighbors = 5,
                   min_dist = 0.1, metric = "euclidean")

umap_df <- data.frame(
  x = mtcar_umap[, 1],
  y = mtcar_umap[, 2],
  engine_type = engine_type,
  transmission = transmission,
  cylinders = cylinders,
  car = car_names
)
```

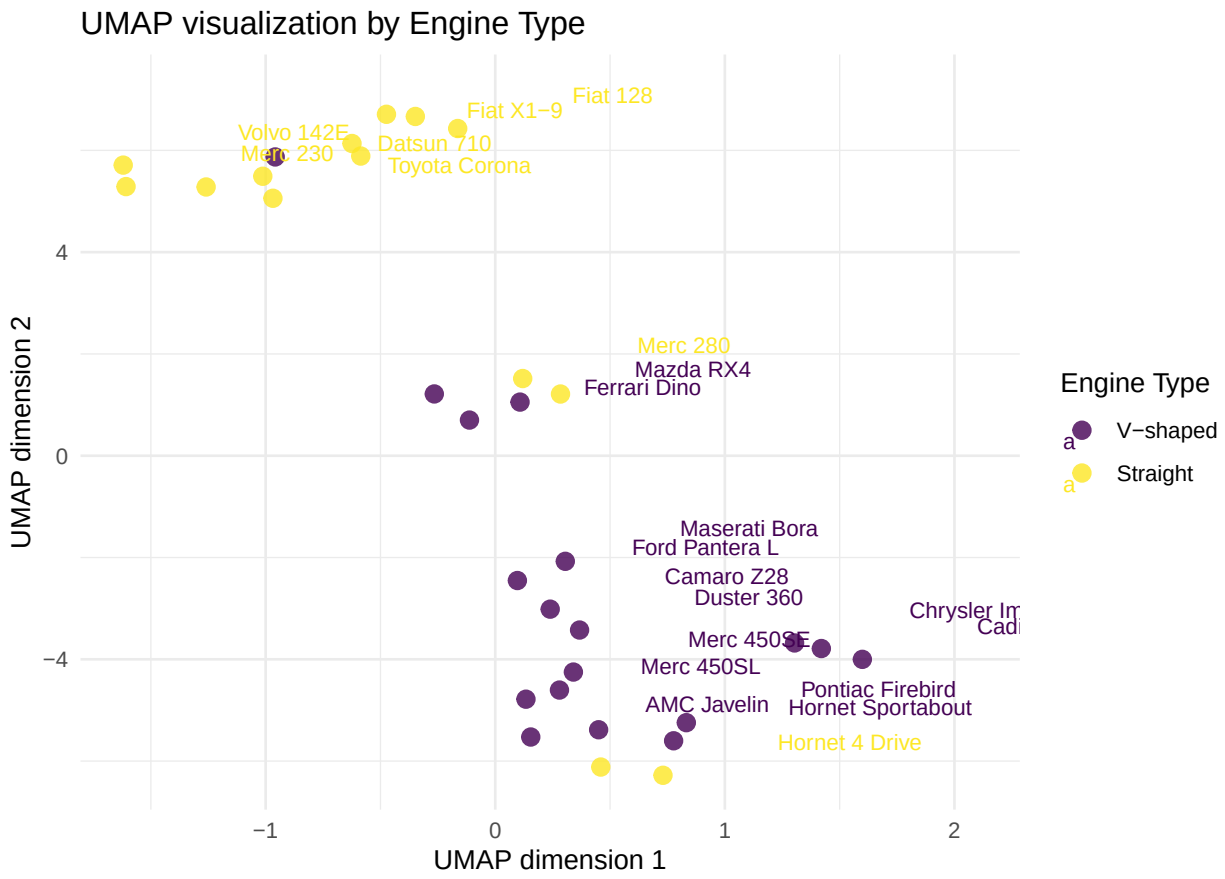
```r
# Xem trước dữ liệu
head(umap_df)
```

```
##                           x         y engine_type transmission cylinders
## Mazda RX4         0.1079102  1.053833    V-shaped       Manual         6
## Mazda RX4 Wag    -0.2654800  1.213801    V-shaped       Manual         6
## Datsun 710       -1.0124921  5.491428    Straight       Manual         4
## Hornet 4 Drive    0.7305642 -6.279060    Straight    Automatic         6
## Hornet Sportabout 0.7768756 -5.601020    V-shaped    Automatic         8
## Valiant           0.4595929 -6.116867    Straight    Automatic         6
##                               car
## Mazda RX4               Mazda RX4
## Mazda RX4 Wag         Mazda RX4 Wag
## Datsun 710               Datsun 710
## Hornet 4 Drive       Hornet 4 Drive
## Hornet Sportabout Hornet Sportabout
## Valiant                   Valiant
```

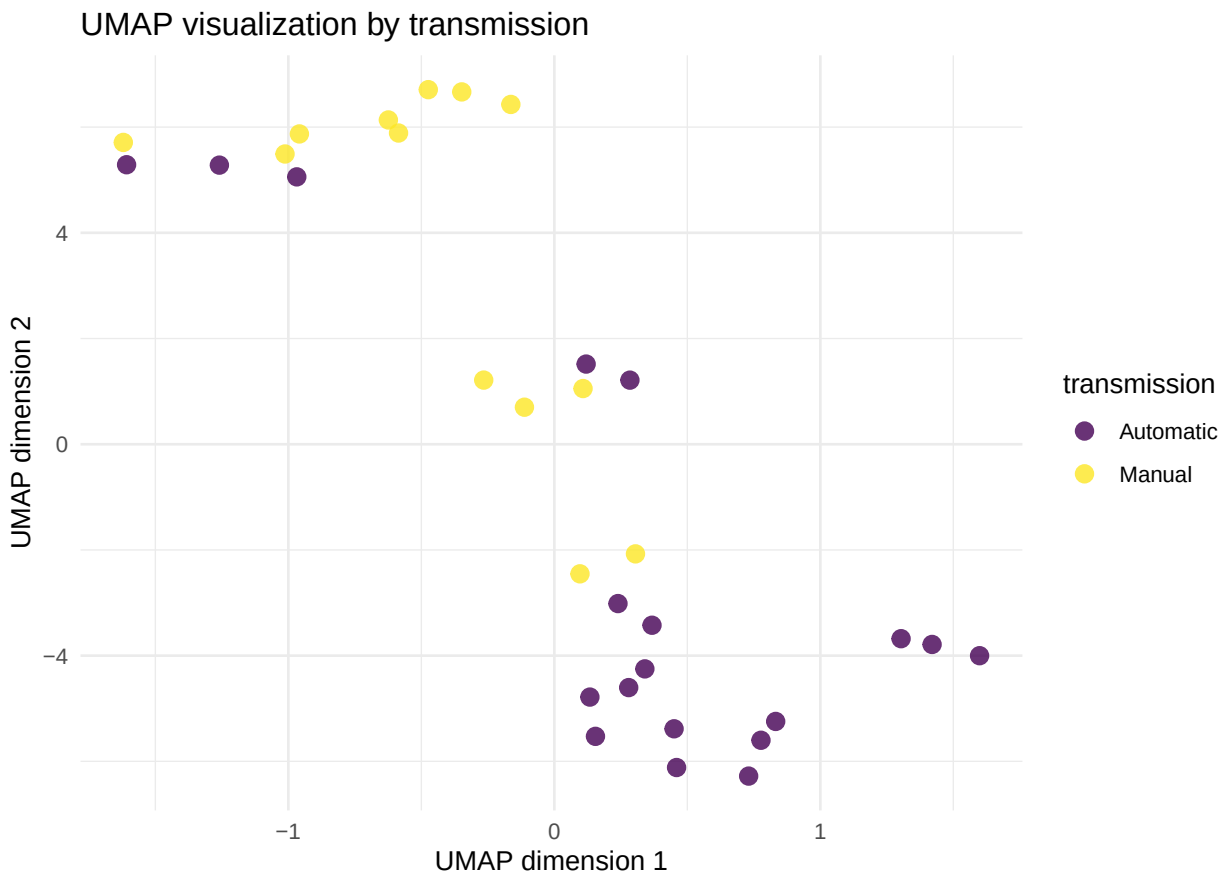
2.6 Trực quan hóa UMAP

```r
# Trực quan hóa theo loại động cơ
p1 <- ggplot(umap_df, aes(x = x, y = y, color = engine_type)) +
  geom_point(size = 3, alpha = 0.8) +
  geom_text(aes(label = car), hjust = 0, vjust = 0, size = 3, nudge_x = 0.5, nudge_y = 0.5, check_overla
  scale_color_viridis_d() +
  labs(title = "UMAP visualization by Engine Type",
       x = "UMAP dimension 1",
       y = "UMAP dimension 2",
       color = "Engine Type") +
  theme_minimal()


p2 <- ggplot(umap_df, aes(x = x, y = y, color = transmission)) +
  geom_point(size = 3, alpha = 0.8) +
  scale_color_viridis_d() +
  labs(title = "UMAP visualization by transmission",
       x = "UMAP dimension 1",
       y = "UMAP dimension 2",
       color = "transmission") +
  theme_minimal()


p3 <- ggplot(umap_df, aes(x = x, y = y, color = cylinders)) +
  geom_point(size = 3, alpha = 0.8) +
  scale_color_viridis_d() +
  labs(title = "UMAP visualization by cylinders",
       x = "UMAP dimension 1",
       y = "UMAP dimension 2",
       color = "cylinders") +
  theme_minimal()
```

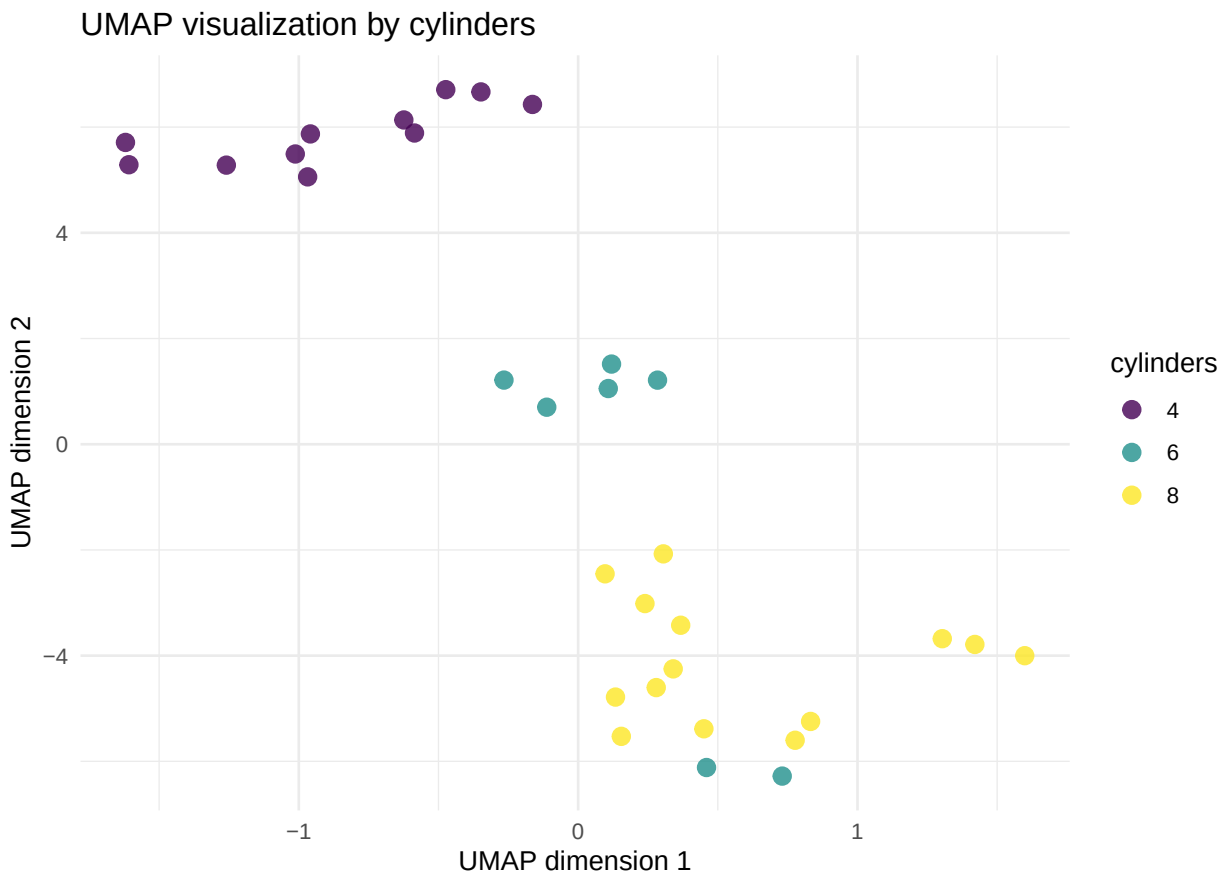
```
p1
```


UMAP visualization by Engine Type
Fiat 128
Fiat X1−9
Volvo 142E
Datsun 710
Merc 230
Toyota Corona
Merc 280
Mazda RX4
Ferrari Dino
Maserati Bora
Ford Pantera L
Camaro Z28
Duster 360
Chrysler Im
Cad
Merc 450SE
Merc 450SL
Pontiac Firebird
Hornet Sportabout
AMC Javelin
Hornet 4 Drive
UMAP dimension 2
UMAP dimension 1
4
0
−4
−1
0
1
2
Engine Type
V−shaped
Straight

```
p2
```


UMAP visualization by transmission
UMAP dimension 2
4
0
−4
−1
0
1
UMAP dimension 1
transmission
Automatic
Manual

```
p3
```

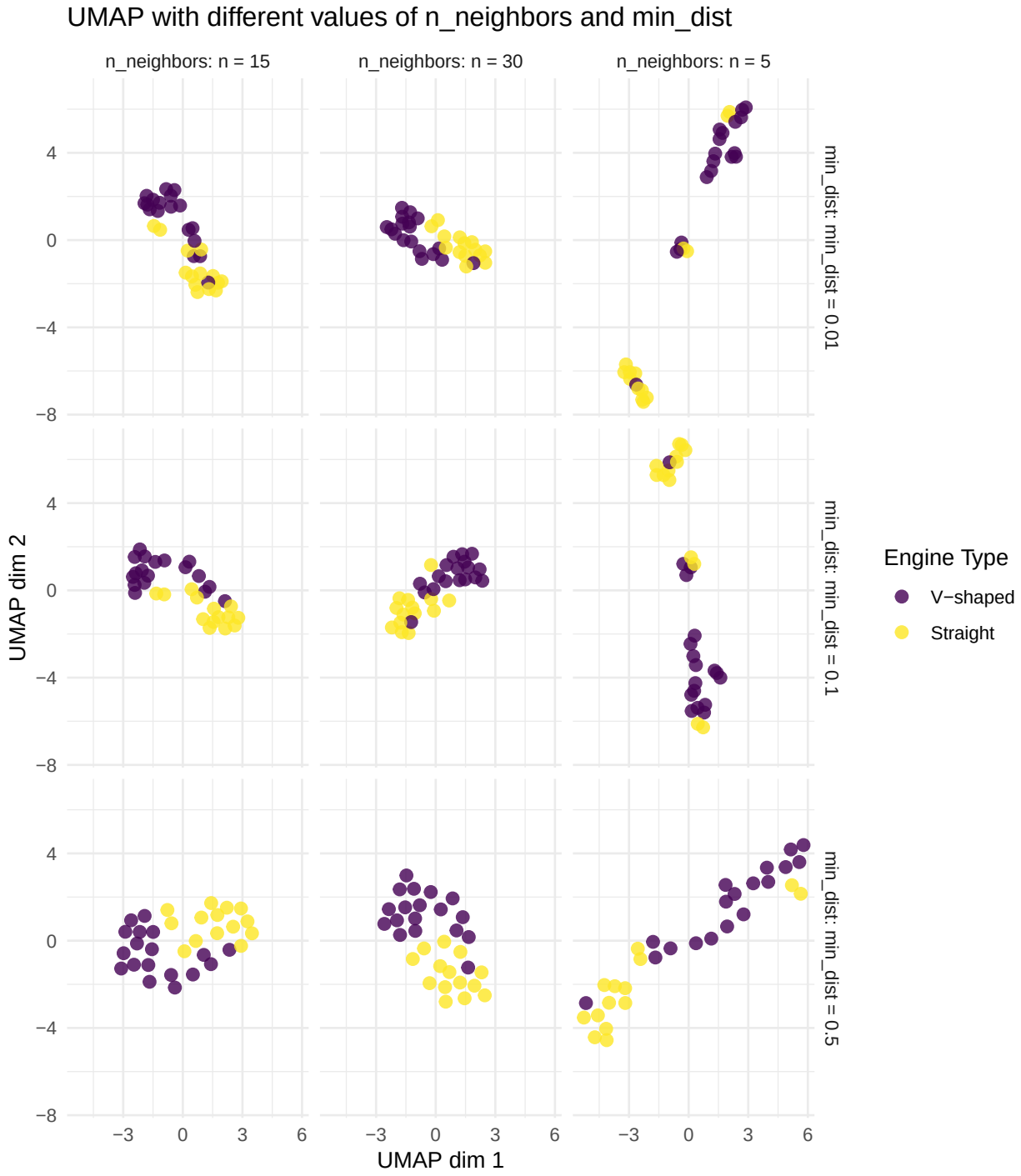


Nhận xét: - UMAP đã tạo ra một biểu diễn hai chiều từ dữ liệu mtcars, trong đó các xe có đặc điểm kỹ thuật tương đồng thường được phân bố gần nhau trên mặt phẳng. - Kết quả cho thấy có sự phân tách khá rõ ràng giữa các loại động cơ (engine_type), kiểu hộp số (transmission) và số xi-lanh (cylinders). - Quan sát cho thấy các xe sử dụng động cơ V-shaped có xu hướng tạo thành một nhóm riêng biệt, tương tự với các xe có cùng số xi-lanh.

2.7 Thực hiện UMAP với các tham số khác nhau

```
library(uwot)
library(dplyr)
library(ggplot2)

run_umap <- function(n_neighbors_val, min_dist_val) {
  set.seed(42)
  result <- umap(mtcars_scaled,
                 n_neighbors = n_neighbors_val,
                 min_dist = min_dist_val,
                 metric = "euclidean")

  data.frame(
    x = result[, 1],
    y = result[, 2],
    engine_type = mtcars_labeled$engine_type,
    transmission = mtcars_labeled$transmission,
    cylinders = mtcars_labeled$cylinders,
```

```
    car = rownames(mtcars),
    n_neighbors = paste0("n = ", n_neighbors_val),
    min_dist = paste0("min_dist = ", min_dist_val)
  )
}

neighbors_values <- c(5, 15, 30)
min_dist_values <- c(0.01, 0.1, 0.5)

params_grid <- expand.grid(n_neighbors = neighbors_values,
                           min_dist = min_dist_values)

umap_all <- bind_rows(
  mapply(function(n, d) run_umap(n, d),
         params_grid$n_neighbors, params_grid$min_dist,
         SIMPLIFY = FALSE)
)

ggplot(umap_all, aes(x = x, y = y, color = engine_type)) +
  geom_point(size = 2.5, alpha = 0.8) +
  scale_color_viridis_d() +
  facet_grid(min_dist ~ n_neighbors, labeller = label_both) +
  labs(title = "UMAP with different values of n_neighbors and min_dist",
       x = "UMAP dim 1",
       y = "UMAP dim 2",
       color = "Engine Type") +
  theme_minimal()
```



Nhận xét: UMAP cho thấy mức độ tách biệt giữa hai loại động cơ phụ thuộc vào giá trị n_neighbors và min_dist. Khi n_neighbors = 5 và min_dist = 0.1, hai nhóm V-shaped và Straight được phân tách rõ ràng nhất. Khi tăng n_neighbors, các cụm dần hòa trộn, đặc biệt rõ ở n = 30. Giá trị min_dist càng lớn thì cụm càng bị dàn trải, làm giảm độ rõ nét. Tổng thể, tổ hợp n_neighbors = 5, min_dist = 0.1 cho kết quả trực quan tốt nhất.

3 so sánh UMAP với PCA

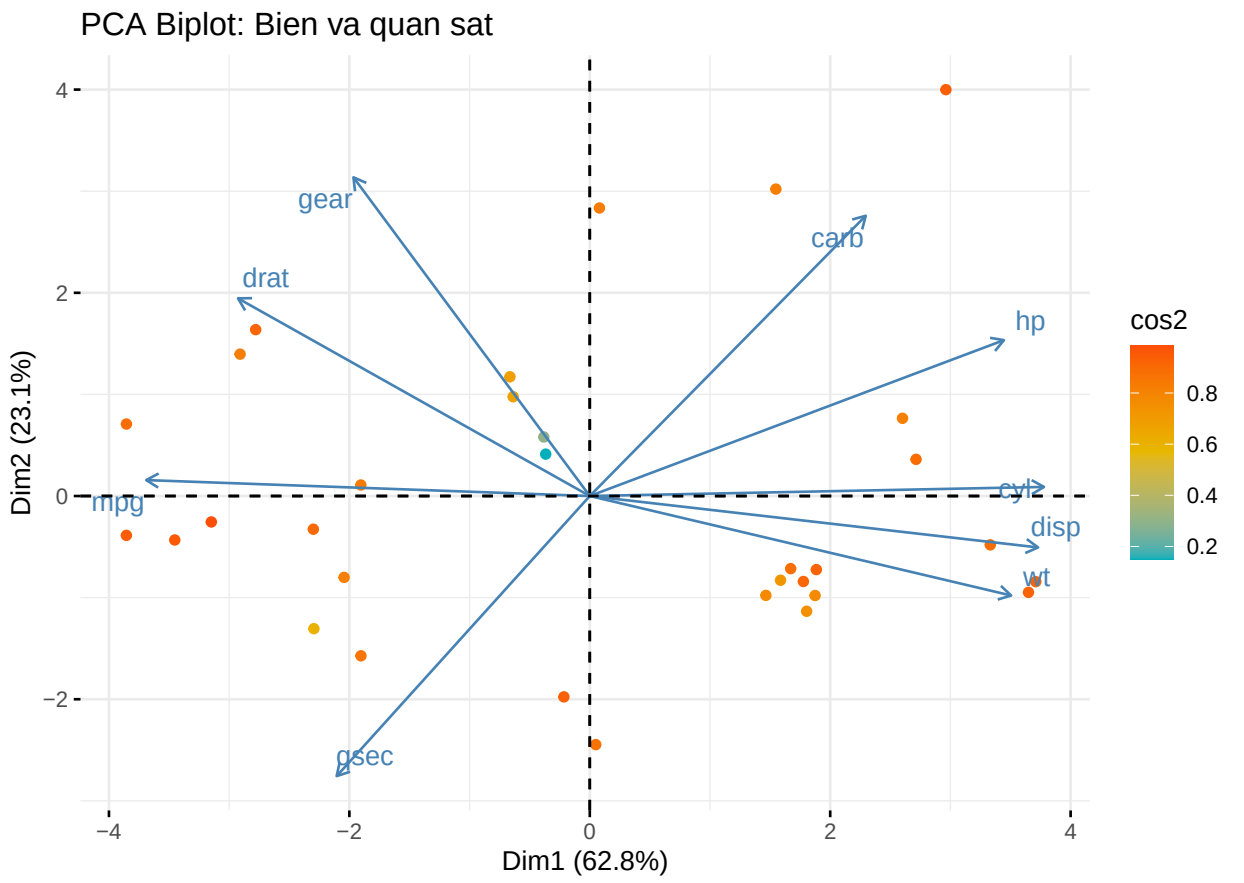
3.1 Thực hiện PCA

```
# Thực hiện PCA trên cùng bộ dữ liệu mtcars
pca_result <- prcomp(mtcars_scaled, center = TRUE, scale. = TRUE)

# Tạo dataframe với kết quả PCA và nhãn
pca_df <- data.frame(
  x = pca_result$x[, 1],
  y = pca_result$x[, 2],
  engine_type = engine_type,
  transmission = transmission,
  cylinders = cylinders,
  car = car_names
)

# Xem tóm tắt kết quả PCA
summary(pca_result)
```

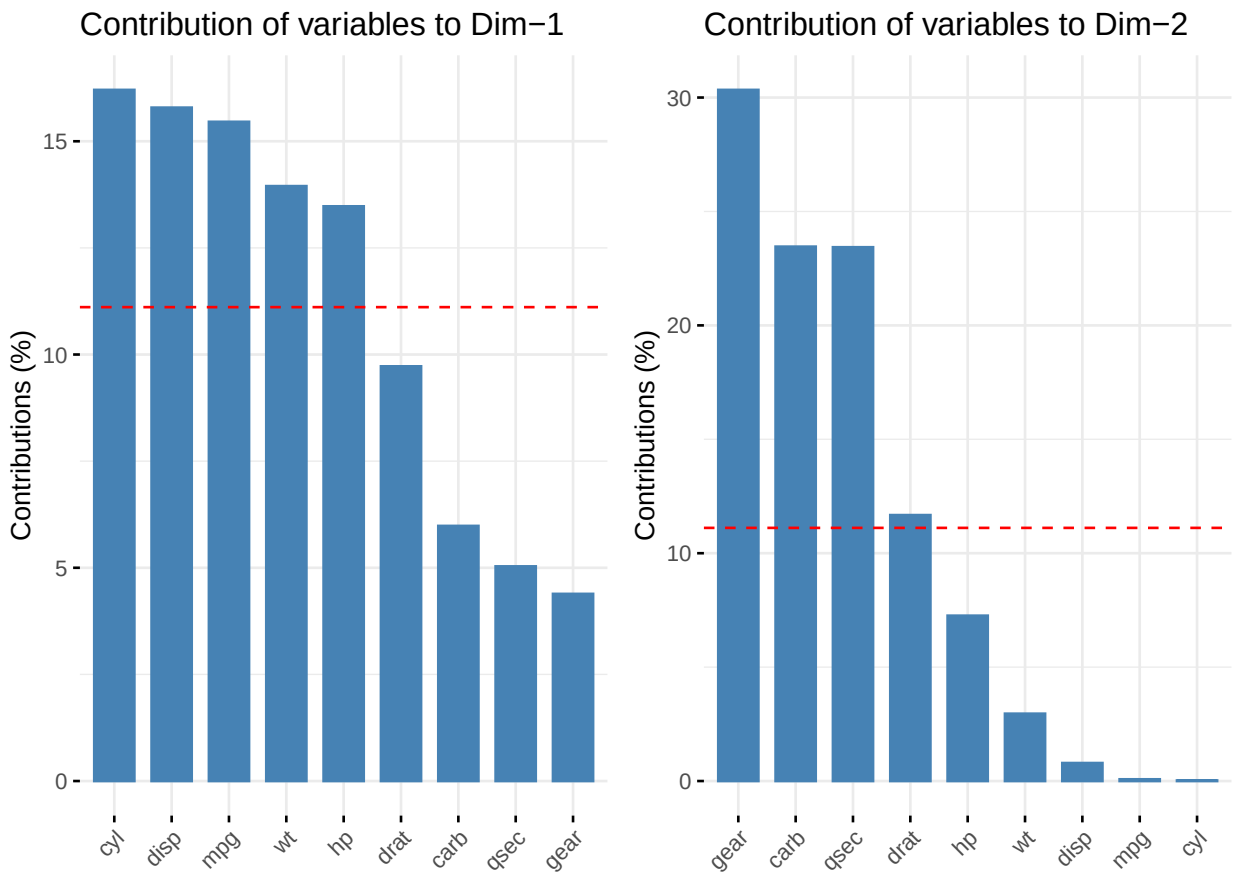
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.3782 1.4429 0.71008 0.51481 0.42797 0.35184 0.32413
## Proportion of Variance 0.6284 0.2313 0.05602 0.02945 0.02035 0.01375 0.01167
## Cumulative Proportion  0.6284 0.8598 0.91581 0.94525 0.96560 0.97936 0.99103
##                           PC8     PC9
## Standard deviation     0.2419 0.14896
## Proportion of Variance 0.0065 0.00247
## Cumulative Proportion  0.9975 1.00000
```

3.2 Phân tích các thành phần chính

```
# Biplot phân tích các biến đóng góp vào PCA
fviz_pca_biplot(pca_result,
                label = "var",
                col.ind = "cos2",
                gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
                repel = TRUE,
                title = "PCA Biplot: Bien va quan sat")
```



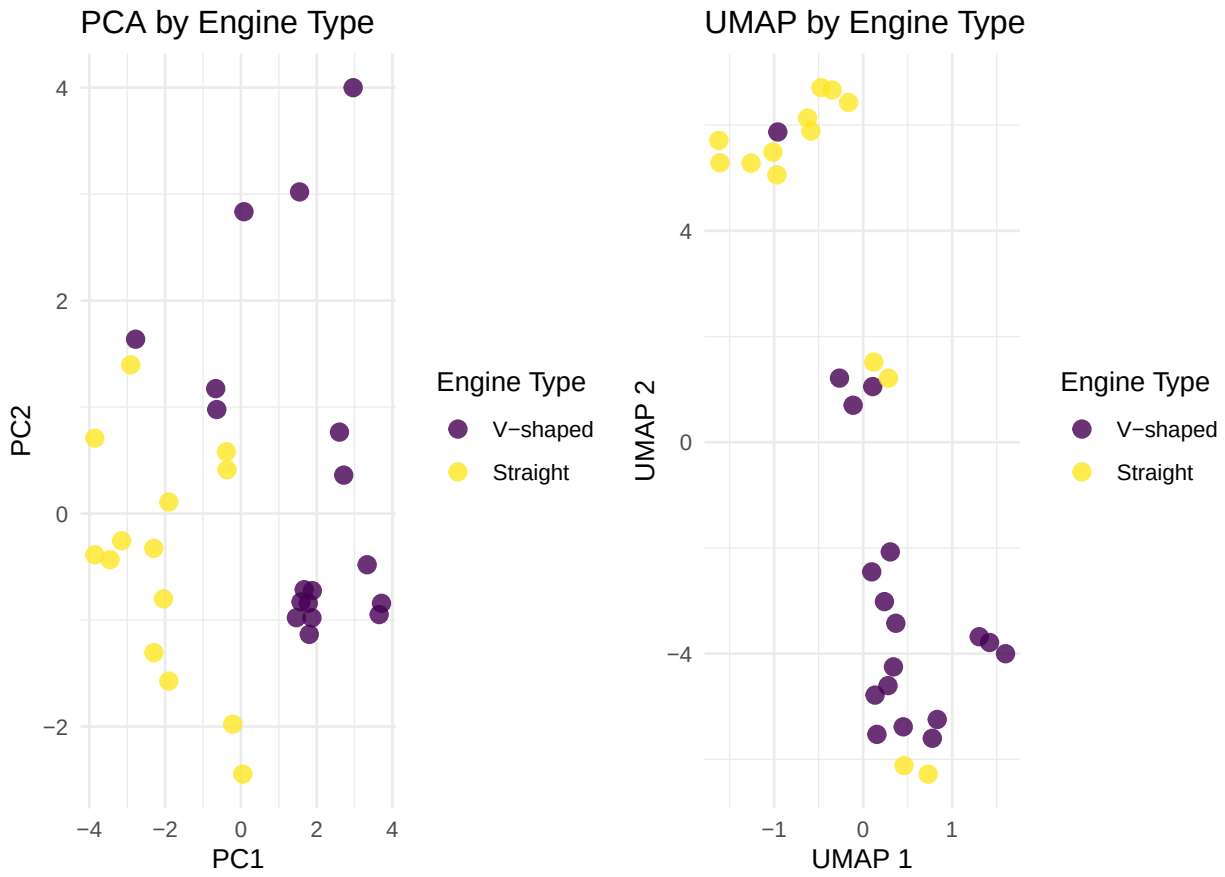

```
# Phân tích đóng góp của các biến vào PC1 và PC2
p1 <- fviz_contrib(pca_result, choice = "var", axes = 1, top = 10)
p2 <- fviz_contrib(pca_result, choice = "var", axes = 2, top = 10)
gridExtra::grid.arrange(p1, p2, ncol = 2)
```



3.3 So sánh trực quan giữa PCA và UMAP

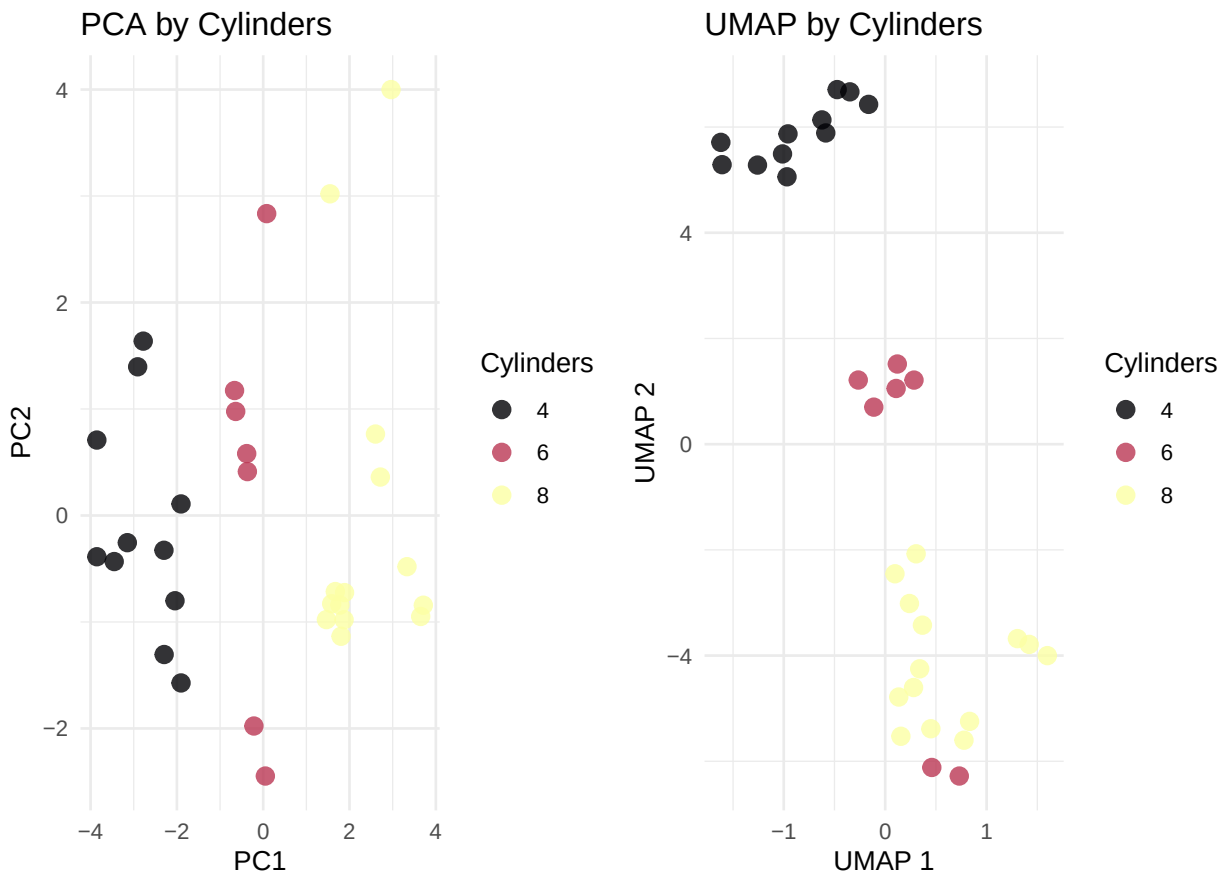
```
# Vẽ biểu đồ PCA theo kiểu động cơ
pca_plot1 <- ggplot(pca_df, aes(x = x, y = y, color = engine_type)) +
  geom_point(size = 3, alpha = 0.8) +
  scale_color_viridis_d() +
  labs(title = "PCA by Engine Type",
       x = "PC1",
       y = "PC2",
       color = "Engine Type") +
  theme_minimal()

# Vẽ biểu đồ t-SNE theo kiểu động cơ
umap_plot1 <- ggplot(umap_df, aes(x = x, y = y, color = engine_type)) +
  geom_point(size = 3, alpha = 0.8) +
  scale_color_viridis_d() +
  labs(title = "UMAP by Engine Type",
       x = "UMAP 1",
       y = "UMAP 2",
       color = "Engine Type") +
  theme_minimal()
```

```
# Hiển thị so sánh theo kiểu động cơ
gridExtra::grid.arrange(pca_plot1, umap_plot1, ncol = 2)
```




```
# Vẽ biểu đồ PCA theo số xi-lanh
pca_plot2 <- ggplot(pca_df, aes(x = x, y = y, color = cylinders)) +
  geom_point(size = 3, alpha = 0.8) +
  scale_color_viridis_d(option = "inferno") +
  labs(title = "PCA by Cylinders",
       x = "PC1",
       y = "PC2",
       color = "Cylinders") +
  theme_minimal()

# Vẽ biểu đồ t-SNE theo số xi-lanh
umap_plot2 <- ggplot(umap_df, aes(x = x, y = y, color = cylinders)) +
  geom_point(size = 3, alpha = 0.8) +
  scale_color_viridis_d(option = "inferno") +
  labs(title = "UMAP by Cylinders",
       x = "UMAP 1",
       y = "UMAP 2",
       color = "Cylinders") +
  theme_minimal()

# Hiển thị so sánh theo số xi-lanh
```

```
gridExtra::grid.arrange(pca_plot2, umap_plot2, ncol = 2)
```



Nhận xét khi so sánh PCA và UMAP trên bộ dữ liệu mtcars:

- PCA cho thấy có xu hướng tách nhóm giữa động cơ V-shaped và Straight, nhưng các điểm vẫn còn xen kẽ, cụm chưa thực sự rõ ràng.
- UMAP thể hiện rõ ràng hơn: hai nhóm động cơ được phân tách tốt hơn, cụm không bị lẫn và có khoảng cách ổn định. Kết luận : UMAP cho kết quả trực quan hóa cụm xi-lanh rõ ràng và sắc nét hơn PCA.

3.4 Phân tích bộ dữ liệu USArrests với UMAP

```
# Tải bộ dữ liệu USArrests
data(USArrests)

# Xem cấu trúc và thông tin cơ bản
str(USArrests)
```

```
## 'data.frame':    50 obs. of  4 variables:
##  $ Murder  : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...
##  $ Assault : int  236 263 294 190 276 204 110 238 335 211 ...
##  $ UrbanPop: int  58 48 80 50 91 78 77 72 80 60 ...
##  $ Rape    : num  21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 31.9 25.8 ...
```

```r
head(USArrests)
```

```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
## Colorado      7.9     204       78 38.7
```

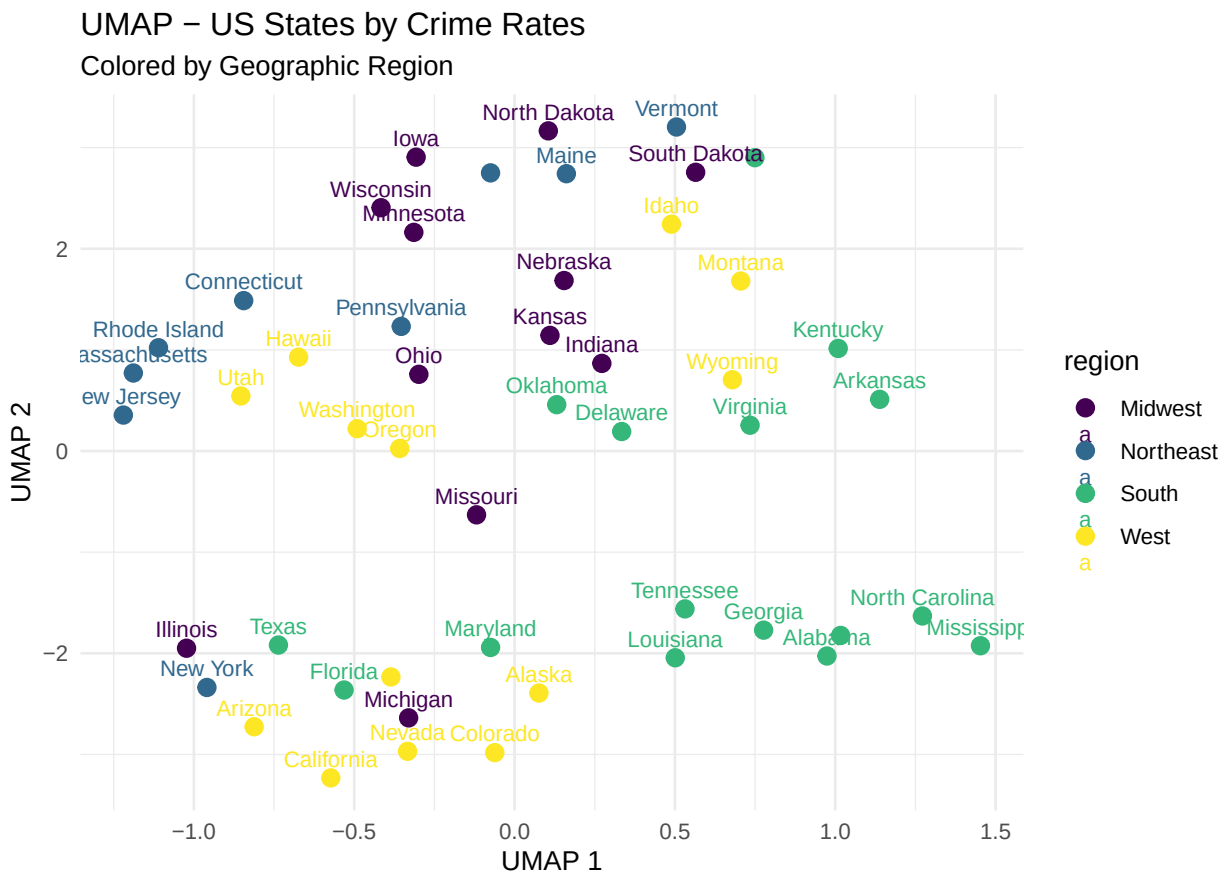
```r
# Tóm tắt thống kê
summary(USArrests)
```

```
##      Murder          Assault         UrbanPop          Rape
##  Min.   : 0.800   Min.   : 45.0   Min.   :32.00   Min.   : 7.30
##  1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:54.50   1st Qu.:15.07
##  Median : 7.250   Median :159.0   Median :66.00   Median :20.10
##  Mean   : 7.788   Mean   :170.8   Mean   :65.54   Mean   :21.23
##  3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:77.75   3rd Qu.:26.18
##  Max.   :17.400   Max.   :337.0   Max.   :91.00   Max.   :46.00
```

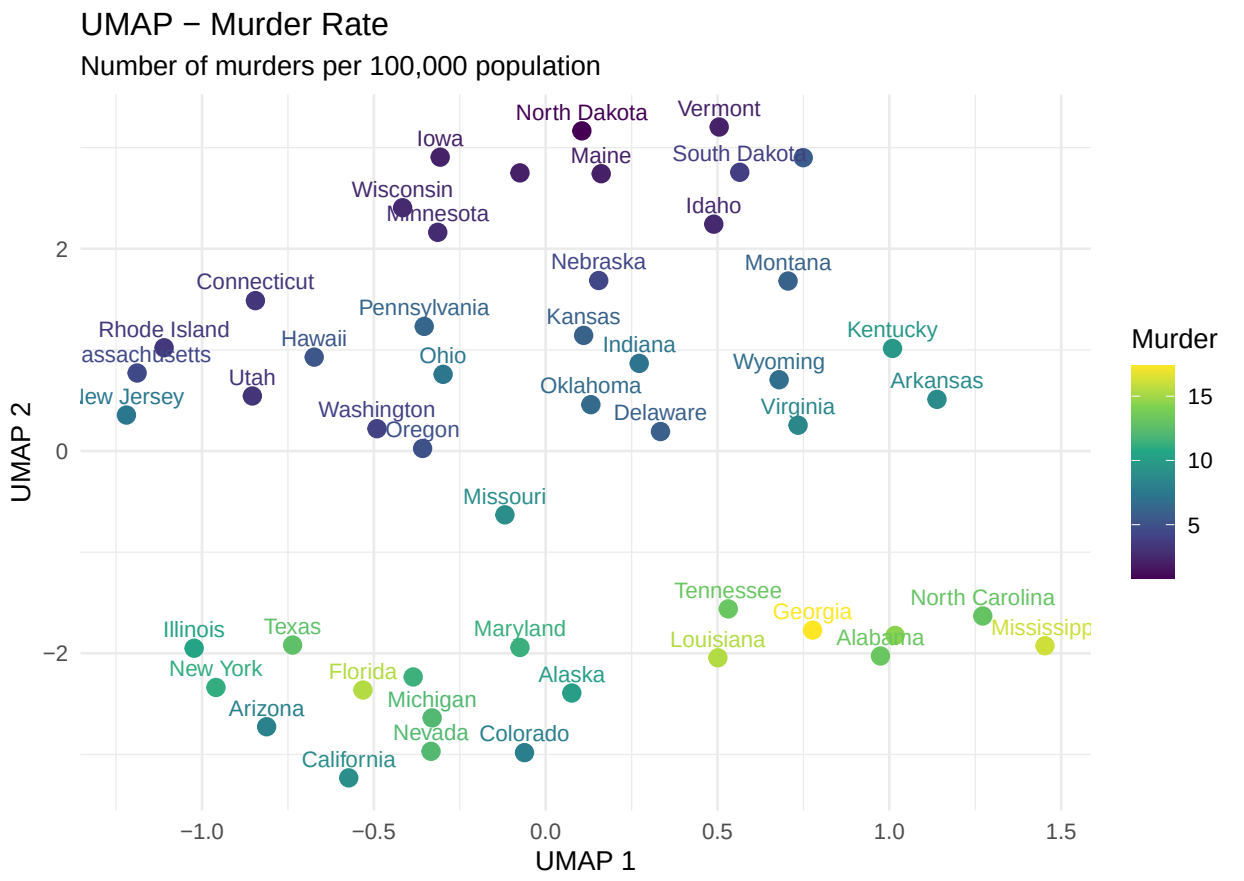
Bộ dữ liệu USArrests chứa thông tin về tỷ lệ tội phạm cho 50 bang của Hoa Kỳ vào năm 1973 với các biến: - Murder: Số vụ giết người trên 100,000 dân - Assault: Số vụ hành hung trên 100,000 dân - UrbanPop: Tỷ lệ dân số đô thị (%) - Rape: Số vụ hiếp dâm trên 100,000 dân

```r
# Chuẩn bị dữ liệu
state_names <- rownames(USArrests)
arrests_data <- USArrests

# Chuẩn hóa dữ liệu
arrests_scaled <- scale(arrests_data)

set.seed(42)
arrests_umap <- umap(arrests_scaled, n_neighbors = 15, min_dist = 0.1)

# Tạo dataframe từ kết quả
arrests_umap_df <- data.frame(
  x = arrests_umap[, 1],
  y = arrests_umap[, 2],
  state = state_names
)

# Ghép dữ liệu gốc
arrests_umap_df <- cbind(arrests_umap_df, arrests_data)

northeast <- c("Maine", "New Hampshire", "Vermont", "Massachusetts", "Rhode Island",
               "Connecticut", "New York", "New Jersey", "Pennsylvania")
midwest <- c("Ohio", "Indiana", "Illinois", "Michigan", "Wisconsin",
             "Minnesota", "Iowa", "Missouri", "North Dakota", "South Dakota",
             "Nebraska", "Kansas")
south <- c("Delaware", "Maryland", "Virginia", "West Virginia", "North Carolina",
           "South Carolina", "Georgia", "Florida", "Kentucky", "Tennessee",
```

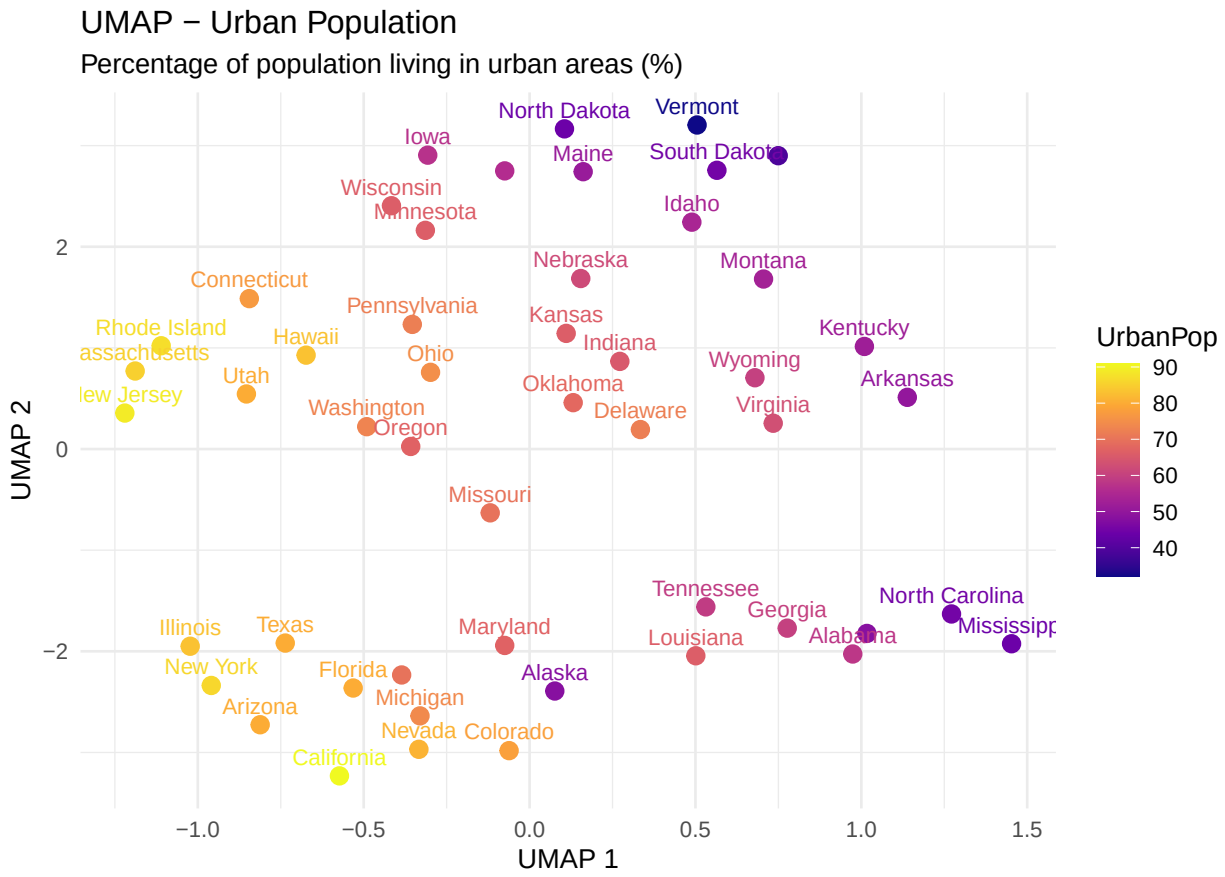
```
           "Alabama", "Mississippi", "Arkansas", "Louisiana", "Oklahoma", "Texas")
west <- c("Montana", "Idaho", "Wyoming", "Colorado", "New Mexico", "Arizona", "Utah",
          "Nevada", "Washington", "Oregon", "California", "Alaska", "Hawaii")

arrests_umap_df$region <- NA
arrests_umap_df$region[arrests_umap_df$state %in% northeast] <- "Northeast"
arrests_umap_df$region[arrests_umap_df$state %in% midwest] <- "Midwest"
arrests_umap_df$region[arrests_umap_df$state %in% south] <- "South"
arrests_umap_df$region[arrests_umap_df$state %in% west] <- "West"
arrests_umap_df$region <- factor(arrests_umap_df$region)

p1 <- ggplot(arrests_umap_df, aes(x = x, y = y, color = region)) +
  geom_point(size = 3) +
  geom_text(aes(label = state), vjust = -0.7, size = 3, check_overlap = TRUE) +
  scale_color_viridis_d() +
  labs(title = "UMAP - US States by Crime Rates",
       subtitle = "Colored by Geographic Region",
       x = "UMAP 1", y = "UMAP 2") +
  theme_minimal()


p2 <- ggplot(arrests_umap_df, aes(x = x, y = y, color = Murder)) +
  geom_point(size = 3) +
  geom_text(aes(label = state), vjust = -0.7, size = 3, check_overlap = TRUE) +
  scale_color_viridis_c() +
  labs(title = "UMAP - Murder Rate",
       subtitle = "Number of murders per 100,000 population",
       x = "UMAP 1", y = "UMAP 2") +
  theme_minimal()


p3 <- ggplot(arrests_umap_df, aes(x = x, y = y, color = UrbanPop)) +
  geom_point(size = 3) +
  geom_text(aes(label = state), vjust = -0.7, size = 3, check_overlap = TRUE) +
  scale_color_viridis_c(option = "plasma") +
  labs(title = "UMAP - Urban Population",
       subtitle = "Percentage of population living in urban areas (%)",
       x = "UMAP 1", y = "UMAP 2") +
  theme_minimal()

p1
```


UMAP − US States by Crime Rates
Colored by Geographic Region
North Dakota
Vermont
Iowa
Maine
South Dakota
Wisconsin
Minnesota
Idaho
Nebraska
Montana
Connecticut
Pennsylvania
Kansas
Rhode Island
Hawaii
Indiana
Kentucky
Ohio
Wyoming
Utah
Oklahoma
Arkansas
Washington
Delaware
Virginia
Oregon
Missouri
Tennessee
North Carolina
Georgia
Illinois
Texas
Maryland
Louisiana
Alabama
Mississipp
New York
Florida
Alaska
Michigan
Arizona
Nevada
Colorado
California
UMAP 2
UMAP 1
region
Midwest
Northeast
South
West

```
p2
```


UMAP − Murder Rate
Number of murders per 100,000 population
North Dakota
Vermont
Iowa
Maine
South Dakota
Wisconsin
Minnesota
Idaho
Nebraska
Montana
Connecticut
Pennsylvania
Kansas
Kentucky
Rhode Island
Hawaii
Indiana
assachusetts
Ohio
Wyoming
Utah
Oklahoma
Arkansas
lew Jersey
Washington
Delaware
Virginia
Oregon
Missouri
Tennessee
North Carolina
Georgia
Illinois
Texas
Maryland
Louisiana
Alabama
Mississipp
New York
Florida
Alaska
Michigan
Arizona
Nevada
Colorado
California
2
0
−2
−1.0
−0.5
0.0
0.5
1.0
1.5
UMAP 2
UMAP 1
Murder
15
10
5

p3



Nhận xét: • UMAP đã biểu diễn dữ liệu USArrests trong không gian 2D, bảo toàn cấu trúc cục bộ và toàn cục tốt hơn PCA. • Các bang trong cùng khu vực địa lý thường gần nhau trong không gian UMAP. • Các bang có tỷ lệ giết người cao hoặc dân số đô thị cao được nhóm lại rõ ràng. • Đây là công cụ mạnh để khám phá và phát hiện mẫu trong dữ liệu xã hội học.

4 Giảm chiều dữ liệu với UMAP cho bài toán thực tế

4.1 Ứng dụng UMAP trong phân tích dữ liệu lớn

Trong phần này, chúng ta sẽ thảo luận cách **UMAP** có thể được áp dụng trong các bài toán thực tế, đặc biệt khi xử lý dữ liệu có số chiều và kích thước lớn.

4.1.1 Quy trình xử lý dữ liệu lớn với UMAP

1. **Làm sạch và chuẩn hóa dữ liệu**
   - Xử lý giá trị thiếu, loại bỏ ngoại lai
   - Chuẩn hóa các biến số để đưa về cùng thang đo
2. **Giảm kích thước dữ liệu (nếu cần)**
   - Với tập dữ liệu rất lớn (hàng triệu mẫu), có thể lấy mẫu đại diện
   - Có thể kết hợp giảm chiều bằng PCA (ví dụ còn 50–100 chiều) trước khi áp dụng UMAP
3. **Áp dụng UMAP**

- Tùy chỉnh các tham số như `n_neighbors` và `min_dist` để điều chỉnh mức độ bảo toàn cấu trúc cục bộ hoặc toàn cục
- Có thể huấn luyện UMAP một lần và áp dụng ánh xạ này cho dữ liệu mới (generalizable)

4. **Giải thích và ứng dụng kết quả**
   - Kết hợp với các thuật toán phân cụm hoặc trực quan hóa
   - Hữu ích trong phân tích khám phá, phát hiện mẫu hoặc bất thường

4.1.2 Ví dụ ứng dụng thực tế của UMAP

1. **Phân tích gene và biểu hiện protein**
   - Giảm chiều dữ liệu gene (vài nghìn chiều) để phân tích cụm tế bào
   - Phát hiện các phân nhóm chức năng hoặc bệnh lý
2. **Xử lý văn bản và ngôn ngữ tự nhiên**
   - Trực quan hóa vector từ hoặc embedding câu văn
   - Hỗ trợ khám phá các nhóm chủ đề tương đồng
3. **Phát hiện gian lận và bất thường**
   - Giảm chiều dữ liệu giao dịch tài chính
   - Nhận diện các điểm bất thường khó thấy trong không gian gốc

4.2 Các khuyến nghị khi sử dụng UMAP

1. **Tiền xử lý dữ liệu:**
   - **Bắt buộc chuẩn hóa** nếu dữ liệu có thang đo khác nhau
   - Không cần loại bỏ trùng lặp như t-SNE
2. **Tham số quan trọng:**
   - `n_neighbors`: kiểm soát độ "cục bộ", thường chọn từ 5–50
   - `min_dist`: kiểm soát khoảng cách tối thiểu giữa các điểm sau khi chiếu xuống không gian thấp
     - Nhỏ → cụm chặt hơn (phù hợp phân cụm)
     - Lớn → cấu trúc lan tỏa hơn (bảo toàn toàn cục)
3. **Khả năng ánh xạ dữ liệu mới:**
   - UMAP có thể **lưu ánh xạ đã học** và **dự đoán cho dữ liệu mới** (`transform()`)
4. **Tính ổn định và tốc độ:**
   - Kết quả thường ổn định hơn t-SNE
   - **Nhanh hơn đáng kể** so với t-SNE, nhất là với dữ liệu lớn

4.3 So sánh UMAP với các phương pháp giảm chiều khác

Phương pháp	Tuyến tính	Bảo toàn cấu trúc	Tốc độ	Khả năng mở rộng	Dễ giải thích	Ứng dụng chính
PCA	Có	Toàn cục	Nhanh	Tốt	Cao	Giảm chiều, khám phá biến chính
t-SNE	Không	Cục bộ	Chậm	Kém	Thấp	Trực quan hóa cụm dữ liệu nhỏ

Phương pháp	Tuyến tính	Bảo toàn cấu trúc	Tốc độ	Khả năng mở rộng	Dễ giải thích	Ứng dụng chính
UMAP	Không	Cục bộ & toàn cục	**Nhanh hơn t-SNE**	**Tốt hơn t-SNE**	Trung bình	Trực quan hóa, phân cụm
LDA	Có	Phân biệt theo nhãn	Nhanh	Tốt	Cao	Phân loại có giám sát
MDS	Tùy	Toàn cục	Trung bình	Trung bình	Trung bình	Phân tích tương đồng
Autoencoder	Không	Tùy thuộc mô hình	Chậm	Tốt	Thấp	Giảm chiều phi tuyến phức tạp

4.4 Một số lưu ý khi sử dụng UMAP

- Tránh hiểu sai: UMAP **không bảo toàn khoảng cách tuyệt đối**, mà bảo toàn **quan hệ hàng xóm** giữa các điểm
- Kết quả phụ thuộc vào tham số `n_neighbors` và `min_dist` → nên thử nghiệm nhiều giá trị
- Có thể dùng cho cả dữ liệu không nhãn và có nhãn (semi-supervised)
- Nếu cần giải thích kết quả kỹ, nên kết hợp PCA để hiểu vai trò của từng biến

4.5 Kết luận

UMAP là công cụ giảm chiều mạnh mẽ, có khả năng:

1. **Bảo toàn cấu trúc dữ liệu cục bộ và toàn cục** tốt hơn t-SNE
2. **Tốc độ xử lý cao**, phù hợp dữ liệu lớn
3. **Có khả năng tổng quát hóa**, áp dụng được cho dữ liệu mới

UMAP không chỉ giúp trực quan hóa dữ liệu đa chiều mà còn hỗ trợ hiệu quả cho các bài toán như phân cụm, khám phá dữ liệu và phát hiện bất thường. Đây là công cụ nên được ưu tiên sử dụng khi làm việc với dữ liệu lớn và phức tạp trong thực tiễn.