SPORTS DATA CAMPUS

MÁSTER EN BIG DATA APLICADO AL SCOUTING EN FÚTBOL
Aplicación de K-Means para la segmentación de perfiles defensivos

SPORTS DATA CAMPUS
UCAM UNIVERSIDAD CATÓLICA DE MURCIA
ENIT INNOVA IT BUSINESS SCHOOL
BIG DATA International Campus

Índice

1. Análisis del código 3

1. Análisis del código

```r

# ----------------------------------------
# 1. Librerías
# ----------------------------------------
library(dplyr)
library(ggplot2)
library(GGally)
library(caret)
library(factoextra)

# ----------------------------------------
# 2. Lectura y limpieza del dataset
# ----------------------------------------
setwd("C:/Users/pepec/Documents/Master/Premaster/PM-Estadística/Modulo3")

# Leer el CSV
df <- read.csv("FBREF_players.csv", sep = ";")

# Filtrar jugadores defensas de LaLiga con al menos 684 minutos jugados
equipos_laliga <- c("Alavés", "Athletic Club", "Atlético Madrid", "Barcelona", "Betis",
                    "Cádiz", "Celta Vigo", "Eibar", "Elche", "Getafe", "Granada",
                    "Huesca", "Levante", "Osasuna", "Real Madrid", "Real Sociedad",
                    "Sevilla", "Valencia", "Valladolid", "Villarreal")

df_defensas <- df %>%
  filter(grepl("DF", Pos),
         Squad %in% equipos_laliga,
         Min >= 684)
```

Tras importar las librerías, establecí el directorio de trabajo mediante la función *setwd()*.

A continuación, leí los archivos del directorio previamente establecido y leemos el archivo .csv que contiene diversas métricas de los jugadores de las cinco grandes ligas que hayan disputado un mínimo de 684 minutos. Seleccioné los equipos de la liga española y después creo el dataframe *df_defensas* que filtra por la posición de defensa.

El resultado es un dataframe limpio y segmentado que servirá como base para el posterior análisis exploratorio y la aplicación de técnicas de clusterización.

```r
# ----------------------------------------
# 3. Selección de métricas
# ----------------------------------------

metricas <- c("Int.90", "Blocks.90", "Recov.90", "AerialW.90",
              "PassesCompleted.90", "KP.90", "PPA.90")

df_metricas <- df_defensas %>%
  select(all_of(metricas)) %>%
  na.omit()
```

Tras filtrar los jugadores por liga y posición, seleccioné un conjunto representativo de KPIs defensivos y de construcción (intercepciones, bloqueos, recuperaciones, duelos aéreos ganados, pases completados, pases clave y pases al área por 90 minutos).

```r
# ---------------------------------------
# 4. Análisis exploratorio
# ---------------------------------------

summary(df_metricas)
ggpairs(df_metricas)
boxplot(scale(df_metricas), main = "Boxplot normalizado de métricas seleccionadas")
```

En esta fase realicé un análisis exploratorio de los datos con el objetivo de comprender la distribución y relación entre las métricas seleccionadas. Primero utilizo la función *summary()* para obtener una visión general de la dispersión y tendencia central de cada variable, identificando posibles valores atípicos o diferencias de escala entre indicadores.

Después, apliqué *ggpairs()* para visualizar las relaciones bivariadas entre las métricas seleccionadas, lo que permite detectar patrones de correlación o redundancia entre variables.

Por último, empleé *boxplot()* sobre los datos normalizados *(scale(df_métricas))* con el fin de comparar gráficamente la variabilidad de las distintas métricas y evaluar su comportamiento antes de aplicar técnicas de clusterización.

```r
# ---------------------------------------
# 5. Normalización de los datos
# ---------------------------------------

preproc <- preProcess(df_metricas, method = c("range"))
df_normalizado <- predict(preproc, df_metricas)

# ---------------------------------------
```

Para evitar que las variables con diferentes escalas afectaran al análisis, apliqué una normalización Min-Max a todas las métricas mediante la función *preProcess()*. Esto permitió homogeneizar los valores en un rango común y garantizar una comparación equitativa entre variables.

```r
# ---------------------------------------
# 6. Número de clusters
# ---------------------------------------

fviz_nbclust(df_normalizado, kmeans, method = "wss") +
  geom_vline(xintercept = 3, linetype = 2)
```

Para determinar el número óptimo de grupos, implementé el método del codo (Elbow Method), calculando la suma de errores cuadráticos dentro de los grupos (WSS). El punto de inflexión identificado en el gráfico correspondió a K = 3, por lo que se estableció la segmentación en tres clusters.

```r
# ----------------------------------------
# 7. K-Means
# ----------------------------------------

set.seed(123)
km <- kmeans(df_normalizado, centers = 3, nstart = 25)

# ----------------------------------------
# ----------------------------------------
# 8. Visualización de clusters
# ----------------------------------------

fviz_cluster(km, data = df_normalizado)
```

Finalmente, apliqué el algoritmo K-Means con 25 inicializaciones distintas para asegurar la convergencia del modelo y evitar resultados dependientes de la inicialización aleatoria. El proceso permitió segmentar a los jugadores en tres grupos funcionales, diferenciados según su perfil de rendimiento defensivo y capacidad de construcción de juego.