Bitácora de Proyecto: Clustering K-means

Descripción General

Este proyecto implementa un análisis de agrupamiento utilizando el método **K-means** en R. En si, el objetivo de este modelo de clustering es agrupar clientes de una empresa de telecomunicaciones en grupos con características similares, basados en sus patrones de uso de llamadas y minutos durante distintos periodos del día. Esto puede ayudar a segmentar clientes para campañas personalizadas, promociones o estrategias de retención. Se utiliza un conjunto de datos llamado `llamadas.csv`, el cual contiene variables como minutos y cantidad de llamadas por franjas horarias.

Detalle de Ejecución

1. Carga de Librerías

Se importan librerías necesarias para el análisis:

- **dplyr**: Manipulación de datos.
- **factoextra**: Visualización de resultados de clustering.
- **cluster**: Funcionalidades para análisis de clusters.
- **gridExtra**: Organización de múltiples gráficos en un único lienzo.

```
library(dplyr)
library(factoextra)
library(cluster)
library(gridExtra)
```

2. Carga y Selección de Datos

Se cargan los datos desde un archivo CSV y se seleccionan las columnas relevantes para el análisis.

Configuración:

- `encoding`: UTF-8.
- `header`: Primera fila contiene nombres de columnas.
- `sep`: Separador de columnas es una coma.

```
df <- read.csv("Files/llamadas.csv", encoding = "UTF-8", header = TRUE, sep = ",",
na.strings = "NA", dec = ".", strip.white = TRUE)

df_calls <- df %>%
  select(day_minutes, day_calls, evening_minutes, evening_calls, night_minutes,
night_calls, intl_minutes, intl_calls)
```

3. División de Datos

Se dividen los datos en un conjunto de **entrenamiento** (70%) y otro de **prueba** (30%) para garantizar la validez de los resultados. Esto se logra seleccionando aleatoriamente índices.

```
set.seed(123)
train_indices <- sample(1:nrow(df_calls), size = 0.7 * nrow(df_calls))
train_data <- df_calls[train_indices, ]
test_data <- df_calls[-train_indices, ]
```

4. Escalado de Variables

Dado que K-means es sensible a la magnitud de las variables, se normalizan las características de los datos.

```
train_data_scaled <- scale(train_data)
test_data_scaled <- scale(test_data)
```

5. Determinación del Número Óptimo de Clusters

Se utilizan dos métodos principales:

1. **Método del Codo (WSS)**: Evalúa la inercia intra-grupos.
2. **Método Silhouette**: Mide la cohesión y separación de los clusters.

```
fviz_nbclust(train_data_scaled, kmeans, method = "wss") + labs(title = "Método del
codo")
fviz_nbclust(train_data_scaled, kmeans, method = "silhouette") + labs(title =
"Método Silhouette")
```

6. Entrenamiento del Modelo K-means

El modelo se entrena con 4 clusters. Se inicializan 25 veces los centroides para garantizar resultados estables.

```
kmeans_result <- kmeans(train_data_scaled, centers = 4, nstart = 25)
```

7. Visualización de Resultados

Clusters en Datos de Entrenamiento

Se representan gráficamente los clusters con un área convexa alrededor de cada grupo.

```
fviz_cluster(
  kmeans_result,
```

```
  data = train_data_scaled,
  geom = "point",
  ellipse.type = "convex",
  palette = "jco",
  ggtheme = theme_minimal()
) + labs(title = "Distribución de clientes (Entrenamiento)")
```

Clusters en Datos de Prueba

Análisis similar para el conjunto de prueba.

```
fviz_cluster(
  test_clusters,
  data = test_data_scaled,
  geom = "point",
  ellipse.type = "convex",
  palette = "jco",
  ggtheme = theme_minimal()
) + labs(title = "Distribución de clientes (Prueba)")
```

8. Comparación de Modelos

Se comparan gráficos de K-means con diferentes números de clusters (2 a 5) para evaluar la estructura del modelo.

```
p1 <- fviz_cluster(k2, geom = "point", data = train_data_scaled) + ggtitle("2
Clusters")
p2 <- fviz_cluster(k3, geom = "point", data = train_data_scaled) + ggtitle("3
Clusters")
p3 <- fviz_cluster(k4, geom = "point", data = train_data_scaled) + ggtitle("4
Clusters")
p4 <- fviz_cluster(k5, geom = "point", data = train_data_scaled) + ggtitle("5
Clusters")
grid.arrange(p1, p2, p3, p4, nrow = 2)
```

9. Análisis Específico

Se evalúan las diferencias entre clusters para la variable `day_minutes`.

```
boxplot(train_data_scaled[, "day_minutes"] ~ kmeans_result$cluster, main =
"Diferencias por cluster en Day Minutes")
```

Conclusión

El análisis K-means permitió segmentar clientes en 4 grupos principales basados en patrones de llamadas. Los métodos utilizados garantizan robustez en el modelo y claridad en los resultados.