Informe Implementacion K-means con distintas distancias

Juan Jose Valencia Montes - Maycol Anderson Ruales Tobar - Matias Silva

Octubre 2024

1 Conjunto de Datos

Exportamos el conjunto de datos a CSV para utilizarlos para un análisis posterior. Se importan librerías y se cargan los datos desde una URL en formato CSV, revisando su estructura. Se obtienen estadísticas resumidas y detalles sobre valores nulos, junto con la forma del conjunto de datos (número de filas y columnas). Mediante una gráfica de barras, visualizamos fácilmente las 10 ligas con mayor cantidad de jugadores. Se crea un "overall" para observar su distribución en el conjunto de datos, lo cual nos permite identificar las clasificaciones de los jugadores.

2 Visualización de la Distribución

Se agrupan los jugadores según su posición y se calcula la mediana del "overall", visualizándola con un diagrama de cajas (*boxplot*). Se realizan múltiples *boxplots* para ver la distribución de los atributos físicos y técnicos entre las diferentes posiciones. Se utiliza el método del codo para identificar el número óptimo de *clusters* (k) al observar la suma de cuadrados dentro del *cluster*.

3 Versión 1: Implementación de K-Means con la Librería `sklearn`

3.1 Preparación de los Datos

Se cargan los datos desde la URL y se visualizan las primeras filas. Se obtiene la forma del *dataframe* y se cuentan los valores faltantes de cada columna. Se identifica un grupo de columnas específicas y se imputa cualquier valor faltante con 0.

3.2 Análisis

Se separan las características (X) y la variable (y) del conjunto de datos. Se codifican variables categóricas tales como el pie preferido y las habilidades físicas y técnicas a formato numérico. Se convierten las columnas a tipo numérico y se imputan los valores faltantes con la mediana.

3.3 Clusterización

Se utiliza el método del codo para identificar el número óptimo de *clusters* (k) al observar la suma de cuadrados dentro del *cluster*. Se realiza la clusterización de los datos usando K-Means y se añaden etiquetas de *cluster* al *dataframe*. Se utiliza PCA para reducir la dimensionalidad y visualizar la clusterización en un gráfico 2D. Además, se crea un gráfico 3D para visualizar la clusterización.

4 Versión 2: Implementación de K-Means++ con la Librería `sklearn`

4.1 Preparación de los Datos

Se importan las librerías necesarias para el manejo de datos y su visualización. Se cargan los datos desde la URL, se visualizan las primeras filas y se identifican las dimensiones del *dataframe*. Se identifican y reemplazan los datos faltantes en columnas específicas con 0.

4.2 Análisis y Clusterización

Se separan las características (X) y la etiqueta (y) usando `iloc`. Se codifican las columnas de pie preferido y habilidades físicas en formato numérico. La matriz X se convierte a un *dataframe* y se asegura que todos los datos sean numéricos. Se imputan los valores faltantes utilizando la mediana. El número óptimo de *clusters* se determina mediante el método del codo. Se aplica PCA para reducir a 3 componentes principales y se ajusta el modelo K-Means utilizando el algoritmo K-Means++. Los centroides obtenidos se transforman a través de PCA y se grafican los *clusters* en 2D y 3D.

5 Versión 3: Implementación de K-Means con Distancia Euclidiana

5.1 Preparación de los Datos

Se importan las librerías necesarias para el manejo de datos y su visualización. Se carga el conjunto de datos desde una URL, se manejan los valores faltantes

y se asignan variables independientes (X) y dependientes (y). Se codifican variables categóricas (pie preferido y habilidad de movimientos) a valores numéricos. Se reemplazan los valores faltantes con la mediana de cada columna.

5.2 Análisis y Clusterización

Se aplica PCA para reducir la dimensionalidad de los datos a tres componentes principales. Se usa el método del codo para determinar el número óptimo de *clusters*, graficando la suma de cuadrados dentro del *cluster*. Se calcula la distancia euclidiana, se eligen centroides aleatorios y se agrupan puntos en *clusters*, actualizando los centroides hasta que el cambio es mínimo. Finalmente, se grafican los *clusters* obtenidos tanto en 2D como en 3D.

6 Versión 4: Implementación de K-Means con Distancia de Mahalanobis

6.1 Preparación de los Datos

Se importan las librerías necesarias para el manejo y visualización de los datos. Se cargan los datos desde una URL y se visualizan las primeras tres filas. Se rellenan los datos faltantes con ceros y se asignan las variables X e y. Las variables categóricas se codifican a valores numéricos y se imputan medianas en los datos para manejar los valores nulos.

6.2 Clusterización

Se aplica PCA para reducir los datos a tres componentes principales. El número óptimo de *clusters* se determina mediante el método del codo. Se seleccionan centroides aleatorios y se agrupan los datos, actualizando los centroides según la distancia de Mahalanobis hasta que su posición cambia poco. Se generan gráficos 2D y 3D de los *clusters* obtenidos.

7 Versión 5: Implementación de K-Means con Distancia de Manhattan

7.1 Preparación de los Datos

Se cargan los datos desde la URL y se muestran las primeras filas. Los valores faltantes en ciertas columnas se rellenan con 0. Se extraen las variables independientes (X) y dependientes (y) del *dataframe*. Se codifican las variables categóricas (pie preferido y habilidades físicas y técnicas) y se convierten a formato numérico. Se asegura que todos los datos de la matriz X sean numéricos y se sustituyen los valores faltantes con la mediana de cada columna.

7.2 Clusterización

Se aplica PCA para reducir los datos a tres componentes principales. Se utiliza el método del codo para determinar el número óptimo de *clusters*. Posteriormente, se implementa manualmente el algoritmo K-Means utilizando la distancia de Manhattan. Se grafican los *clusters* y sus centroides en 2D y 3D.