Análisis de Componentes Principales (PCA) de un Conjunto de Datos de blue jays

INFOTEP

Robbyel Elías, Carlos Jerónimo

Abstract

Este documento describe un proceso de análisis de datos utilizando el análisis de componentes principales (PCA) para un conjunto de datos sobre aves. Se aborda la carga de datos, la limpieza de los mismos, la visualización de correlaciones y la determinación de componentes principales que explican el 90% de la varianza del conjunto de datos.

Keywords: Análisis de Componentes Principales, PCA, correlación, análisis bivariado.

1 Introducción

El análisis de componentes principales (PCA) es una técnica estadística utilizada para reducir la dimensionalidad de los datos, extrayendo las componentes más significativas que explican la mayor parte de la varianza. Este análisis es particularmente útil cuando se trabaja con datos de alta dimensionalidad.

2 Cargar y limpiar los datos

El primer paso en el proceso es cargar los datos desde un archivo CSV y luego limpiar los datos, eliminando las columnas no numéricas y manejando los valores faltantes.

```
# Cargar librerías necesarias
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# Cargar datos
file_path = "/content/sample_data/blue_jays.csv"
df = pd.read_csv(file_path)

# Eliminar columnas no numéricas
df_numeric = df.drop(columns=["bird_id", "sex"])
```

```
# Manejo de valores faltantes e infinitos
df_numeric = df_numeric.replace([np.inf, -np.inf], np.nan)  # Reemplazar infinitos por NaN
df_numeric = df_numeric.dropna()  # Eliminar filas con NaN
```

3 Análisis bivariado: Matriz de Correlación

Una parte fundamental del análisis exploratorio es observar las relaciones entre las variables. Para ello, se utiliza un mapa de calor de la matriz de correlación.

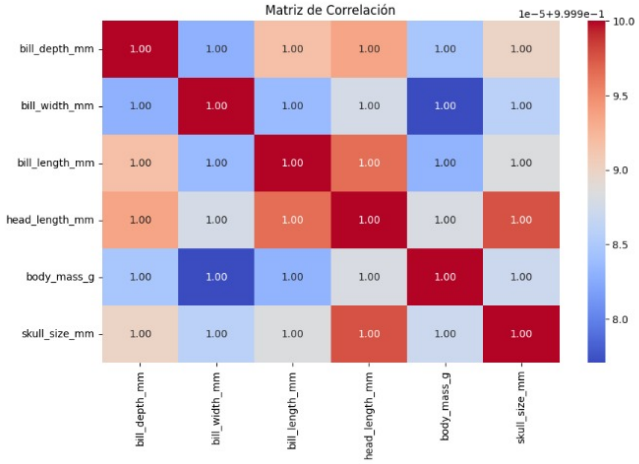

Figure 1: Matriz de correlación del conjunto de datos numéricos.

4 Cálculo de la matriz de covarianza y valores propios

A continuación, calculamos la matriz de covarianza, que describe cómo varían las variables en relación unas con otras. Luego, calculamos los valores y vectores propios que son fundamentales para el PCA.

```
# Calcular matriz de covarianza
cov_matrix = np.cov(df_numeric, rowvar=False)
print("Matriz de Covarianza:")
print(cov_matrix)

# Calcular valores y vectores propios
```

```
eigenvalues, eigenvectors = np.linalg.eig(cov_matrix)
print("Valores propios:")
print(eigenvalues)
print("Vectores propios:")
print(eigenvectors)
```

5 Análisis de Componentes Principales (PCA)

El PCA se realiza utilizando la librería de sklearn, y el número de componentes necesarios para explicar al menos el 90% de la varianza se determina mediante la acumulación de la varianza explicada.

```
# Análisis de Componentes Principales (PCA)
from sklearn.decomposition import PCA

pca = PCA()
pca.fit(df_numeric)
explained_variance = np.cumsum(pca.explained_variance_ratio_)

# Determinar cuántos componentes explican el 90% de la varianza
num_components = np.argmax(explained_variance >= 0.90) + 1
print(f"Número de componentes necesarios para explicar el 90% de la varianza: num_componen
```

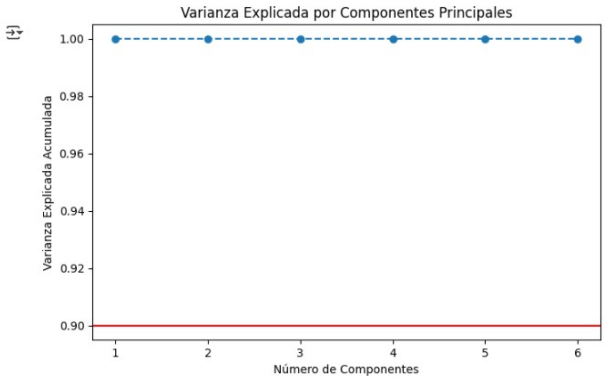

Figure 2: Varianza explicada por componentes principales.

6 Conclusiones

El análisis ha permitido reducir la dimensionalidad del conjunto de datos, identificando los componentes principales que explican el 90% de la varianza. Este

tipo de análisis es útil para simplificar la representación de los datos, facilitando la visualización y la interpretación de patrones subyacentes.

Conflictos de interés. Los autores declaran no tener conflictos de interés.