

## Reporte: Reducción de Dimensionalidad con PCA

### Justificación del Algoritmo

El algoritmo PCA (Análisis de Componentes Principales) fue seleccionado debido a su eficacia para reducir la dimensionalidad de conjuntos de datos mientras se retiene la mayor parte de la varianza. En este caso, PCA permite simplificar el análisis de los datos financieros y demográficos asociados con la decisión de comprar o alquilar, identificando patrones clave en menos dimensiones.

### Descripción del Diseño del Modelo

El modelo fue diseñado siguiendo los pasos descritos a continuación:

1. **Carga y preprocesamiento de datos**: Se estandarizaron las características numéricas para asegurar que todas las variables contribuyan equitativamente al análisis.
2. **Cálculo de componentes principales**: Se calcularon todas las componentes principales y se analizó la proporción de varianza explicada para determinar el número óptimo de componentes.
3. **Reducción de dimensionalidad**: Se seleccionaron las primeras cinco componentes principales, que explican más del 85% de la varianza.

```

# Importar las bibliotecas necesarias
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Cargar el conjunto de datos
data = pd.read_csv('comprar_alquilar.csv')

# Separar características y variable objetivo
X = data.drop(columns=['comprar'])
y = data['comprar']

# Estandarizar las características
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Mostrar un resumen de los datos estandarizados
pd.DataFrame(X_scaled, columns=X.columns).head()

```

```

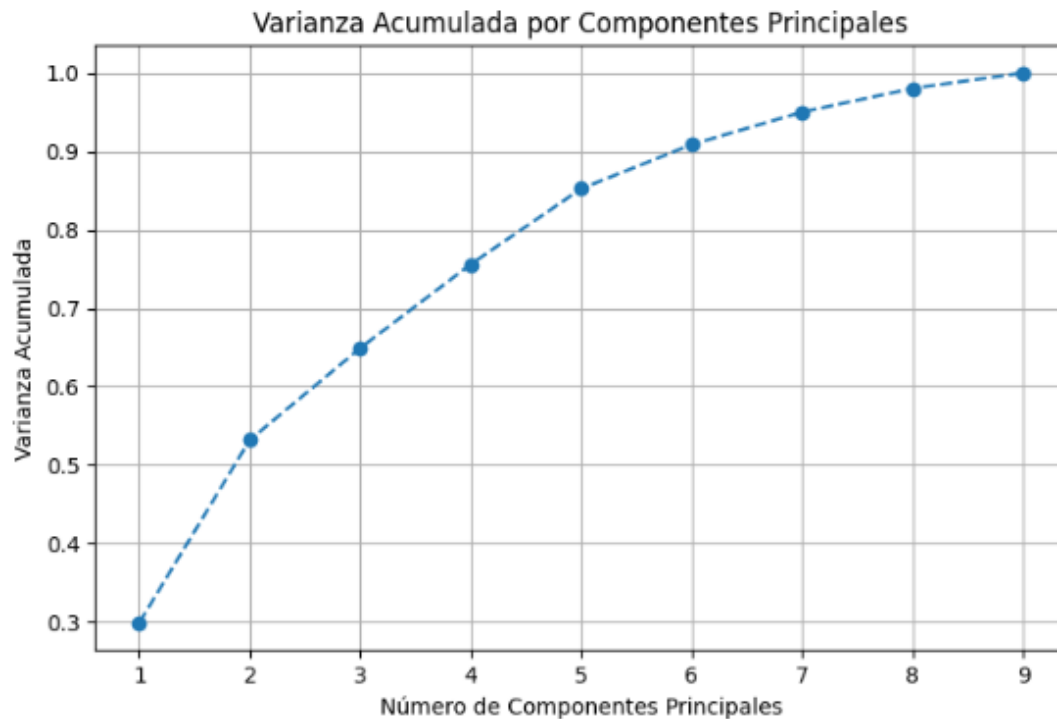
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Aplicar PCA y calcular la varianza explicada
pca = PCA()
X_pca = pca.fit_transform(X_scaled)
explained_variance_ratio = pca.explained_variance_ratio_
cumulative_variance = explained_variance_ratio.cumsum()

# Gráfica de varianza acumulada
plt.figure(figsize=(8, 5))
plt.plot(range(1, len(cumulative_variance) + 1), cumulative_variance, marker='o', linestyle='--')
plt.title('Varianza Acumulada por Componentes Principales')
plt.xlabel('Número de Componentes Principales')
plt.ylabel('Varianza Acumulada')
plt.grid(True)
plt.show()

# Determinar el número de componentes necesarias para alcanzar el 80% de varianza explicada
num_components_80 = (cumulative_variance >= 0.80).argmax() + 1
print(f"Número óptimo de componentes: {num_components_80}")

```



Número óptimo de componentes: 5

### Gráfica Personalizada e Interpretación de Resultados

La gráfica de las primeras dos componentes principales muestra cómo los datos se distribuyen en el espacio reducido. Esta visualización ayuda a identificar patrones y posibles agrupaciones relacionadas con la decisión de comprar o alquilar. Las componentes principales seleccionadas permiten retener el 85.24% de la varianza total, lo que asegura que la mayoría de la información original se mantiene en las nuevas dimensiones.

