Portafolio Análisis

Erika Marlene García Sánchez - A01745158

Grupo: 101

Septiembre 2023

Inteligencia artificial avanzada para la ciencia de datos I

Instituto Tecnológico y de Estudios Superiores de Monterrey

Portafolio Análisis

Justificación selección del dataset

Seleccioné un dataset de las atracciones turísticas de California porque se me hizo interesante analizar la concentración de lugares turísticos en ese Estado. Considero que es un dataset adecuado ya que k-means se usa con datos que no cuenten con su categoría definida ya que es un algoritmo de machine learning no supervisado. Además son valores continuos ya que son coordenadas geográficas. El modelo de K-means está generalizando ya que la división de los datos se está haciendo aleatoriamente, es decir cada vez que se corre el programa la selección de los datos a usar para train, test y validation serán diferentes y aún así el modelo es capaz de hacer una buena predicción de su cluster correspondiente.
Además creo que si se hace un análisis más amplio en el dataset como saber los ingresos que genera cada lugar turístico se puede identificar que zonas causan mayores ingresos y que tipo de atracciones son. Esto con la finalidad de aumentar los ingresos económicos en zonas que no contengan ese tipo de atracciones.

Separación y evaluación del modelo con un conjunto de prueba y un conjunto de validación (Train/Test/Validation)

- Selección de Características: Para el clustering, seleccioné dos características relevantes, "Latitude" y "Longitude", que representan la ubicación geográfica de los lugares turísticos.
- Normalización: Normalicé los datos utilizando la técnica StandardScaler de Scikit-Learn. Esto es esencial para que el algoritmo K-Means funcione correctamente, ya que es sensible a la escala.
- División de Datos:Para separar el conjunto de datos en conjuntos de entrenamiento, prueba y validación, use la función train_test_split de Scikit-Learn. Dividí el 80% de datos en entrenamiento, un 20% en datos de prueba y este último lo volvía a dividir en 25% para datos de validación y el 75% para prueba. Esto es esencial para evaluar adecuadamente el modelo y evitar el sobreajuste.

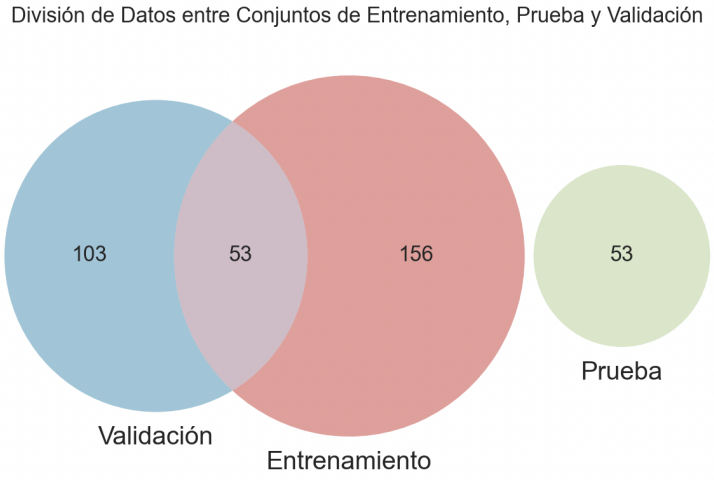

Imagen 1. Captura de pantalla del diagrama de Venn de la separación de datos

```python
# ---------------- PREPROCESAMIENTO ----------------
data = pd.read_csv("./touristic_places_california.csv")

# Selección de las columnas relevantes para el clustering
X = data[['Latitude', 'Longitude']]

# Normalización de los datos utilizando StandardScaler.
# Esto es importante para que K-Means funcione correctamente,
# ya que el algoritmo es sensible a la escala.
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# División en train y test (80% train, 20% test)
train_data, test_data = train_test_split(X_scaled, test_size=0.2,
                                         random_state=None)

# División adicional en train y validation (75% train, 25% validation)
train_data, validation_data = train_test_split(train_data, test_size=0.25,
                                               random_state=None)
print(data.head())
print("\nTrain data:\n", train_data)
print("\nValidation data:\n", validation_data)
print("\nTest data:\n", test_data)
```

```
                                AttractionName  Latitude  Longitude
0                                Blue Lake Museum   40.8810   -123.990
1  Arcata Wastewater Treatment Plant and Wildlife...   40.8583   -124.091
2                     Humboldt Bay Maritime Museum   40.8201   -124.182
3                                  Carson Mansion   40.8027   -124.153
4                                 Sequoia Park Zoo   40.7770   -124.145
```

Imagen 2. Captura de pantalla del proceso de división de los datos así como los primeros 5 datos originales.

```
Train data:
 [[-7.64589322e-01  5.92732038e-01]
 [-7.90710792e-01  4.82154987e-01]
 [-2.32588299e-01 -5.00110057e-01]
 [-1.12544438e+00  1.06243160e+00]
 [-1.21250398e+00  1.04296891e+00]
 [ 5.41663791e-01 -1.94858108e-01]
 [-1.31690859e+00  1.25385060e+00]
 [-6.24728240e-01  4.59953249e-01]
 [ 1.40265130e+00 -5.13267700e-01]
 [ 1.66291504e+00 -7.13501141e-01]
 [ 7.96736280e-01 -1.07630167e+00]
 [-6.00562580e-01  4.51916023e-01]
 [-5.05026910e-01  3.33662023e-01]
 [-7.02259322e-01  8.65438046e-01]
 [-1.26377889e+00  9.61658120e-01]
 [-5.05101892e-01  3.35704308e-01]
```

Imagen 3. Captura de pantalla de los datos asignados a train_data.

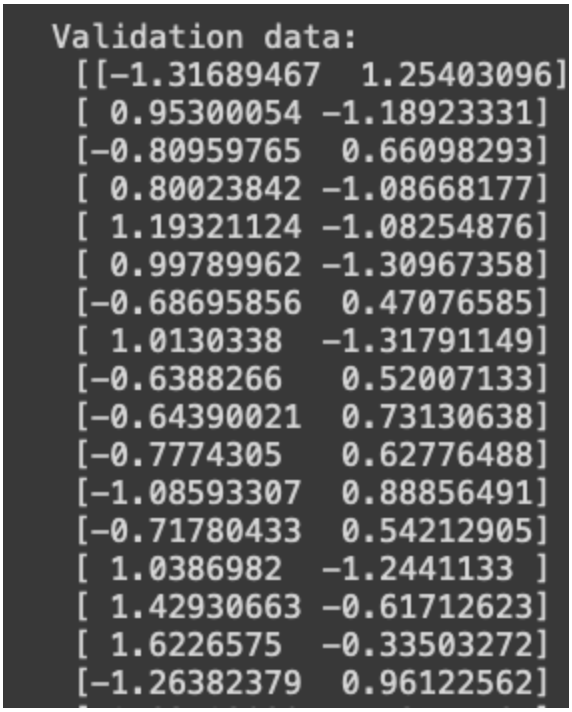

Imagen 4. Captura de pantalla de los datos asignados a validation_data.

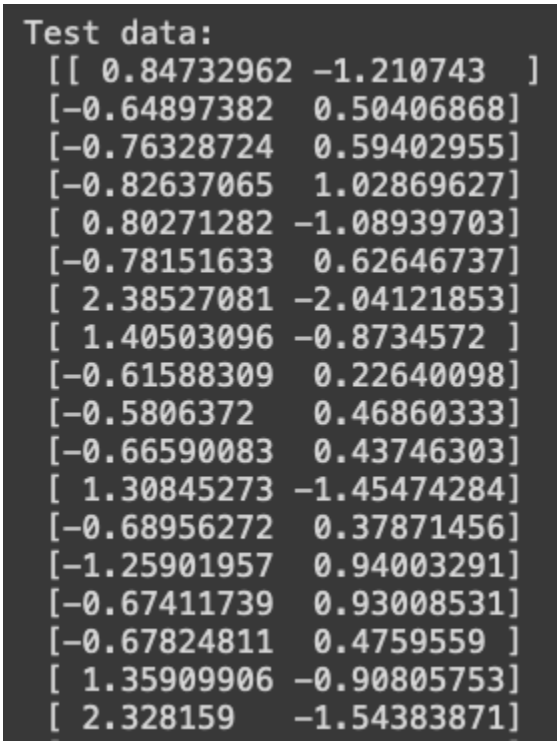

Imagen 5. Captura de pantalla de los datos asignados a test_data.

Diagnóstico y explicación el grado de bias o sesgo: bajo medio alto

- De acuerdo al histograma de predicciones de los tres conjuntos de datos se puede observar que son muy similares en forma y frecuencia (tomando en cuenta que los datos de entrenamiento representan el 80% de los datos totales, los de prueba el 15% y los de validación 5%), esto demuestra una asignación de clusters consistente entre los conjuntos. Por lo cual no hay sesgos en las predicciones.
- Se puede observar que en el gráfico de densidad de predicciones muestra sus distribuciones y como se superponen mostrando que las predicciones en los tres conjuntos son similares.
- Se puede concluir que el grado de sesgo es bajo.

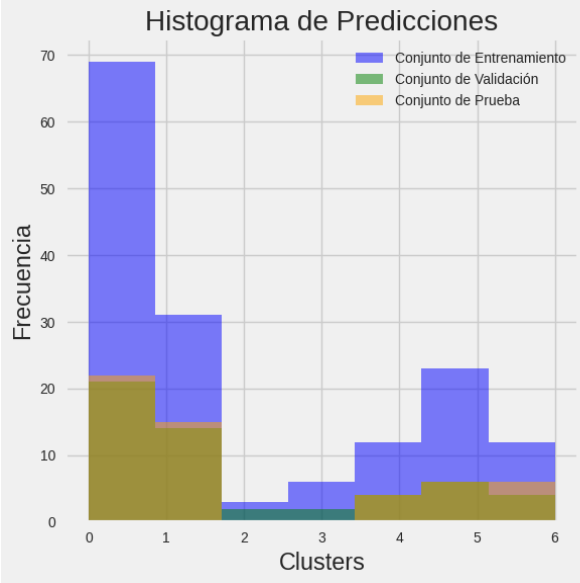

Imagen 6.Histograma que muestra las predicciones que generó para los tres diferentes conjuntos de datos.

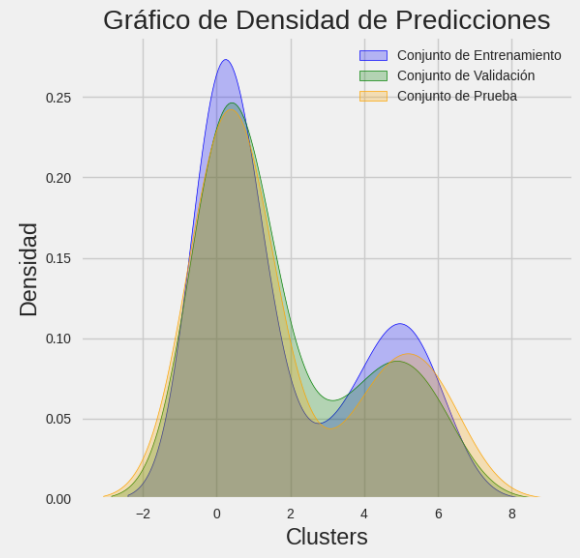

Imagen 7.Gráfico de densidad que muestra las predicciones que generó para los tres diferentes conjuntos de datos.

Diagnóstico y explicación el grado de varianza: bajo medio alto

- Gráfico de Varianza: El coeficiente de Silhouette se utiliza para evaluar la varianza. Se observa que cuando k toma el valor de 7 el coeficiente de Silhouette es de alrededor de 0.60 siendo este cercano a 1 lo que indica que los clusters están bien separados y la asignación de puntos a clusters es apropiada. Además a su derecha cuando k es 8 se observa de igual manera un buen coeficiente de silhouette por lo tanto muestra que el coeficiente es estable antes de una disminución significativa.

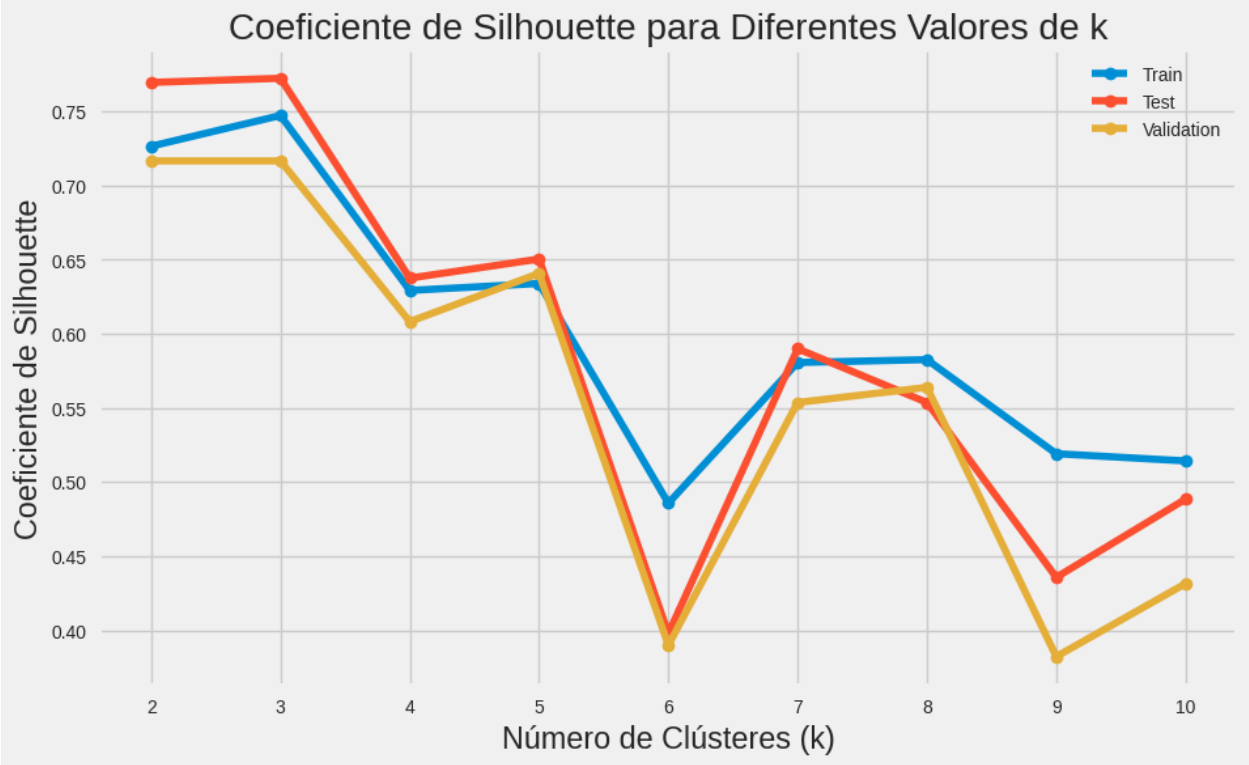

Imagen 8.Gráfico de coeficiente de Silhouette para diferentes número de clusters para los tres diferentes conjuntos de datos.

Diagnóstico y explicación el nivel de ajuste del modelo: underfit, fit, overfit

Debido a que K-means es un modelo no supervisado no cuento con las variables de las etiquetas verdaderas de los clusters y no puedo hacer una curva de aprendizaje. Es por esto que también me apoyaré de los datos obtenidos en el gráfico de coeficiente de Silhouette ya que evalúa la calidad de los clusters. Con esto se puede inferir que el nivel de ajuste del modelo es fit.

Uso de técnicas de regularización o ajuste de parámetros para mejorar el desempeño

- Técnicas de ajuste: En mi código, implementé técnicas de ajuste de parámetros al variar el número de clusters k y calcular el Silhouette Score para seleccionar el valor óptimo de k. Esto me ayudó a mejorar el rendimiento del modelo de clustering. También realice el

método del codo donde me sugería que k tomara el valor de 3 y después de hacer los debidos procesos y visualizar la gráfica de dispersión pude notar que no se encontraban tan separados los datos de otros clusters así que no me pareció una buena k. Después probé que k fuera 5 ya que la gráfica de Silhouette lo sugería y pude ver que mejoraba la clasificación pero aún visualizaba unos datos que no estaban tan cercanos a otros datos de su cluster así que probé con otro valor sugerido en la gráfica de Silhouette que es 7 y pude notar que ahora los datos ya están clasificados de manera correcta.

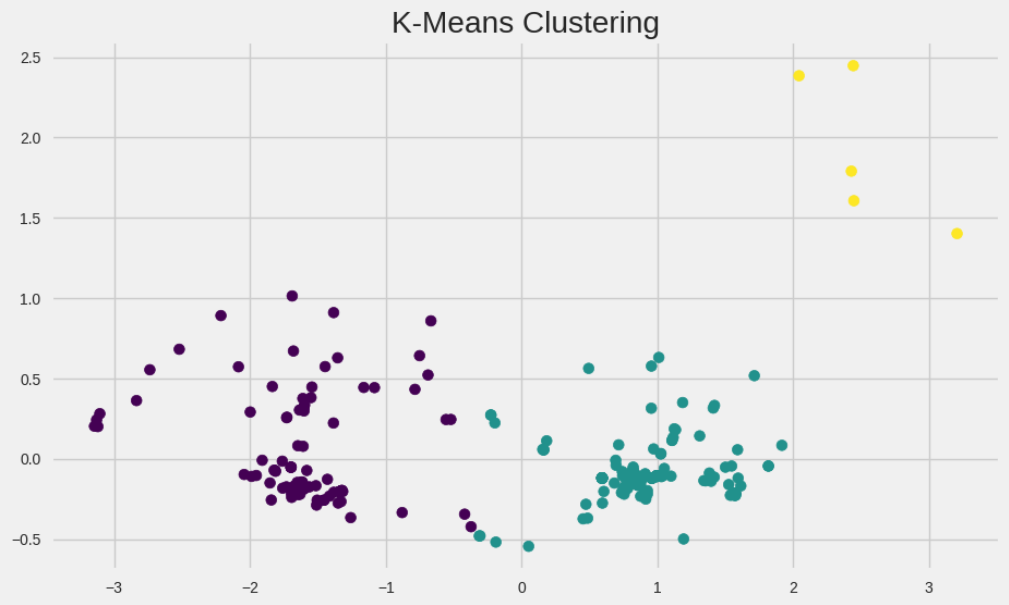

Imagen 9. Gráfica de dispersión que muestra cuando k= 3,
que es el valor sugerido en el análisis del codo.

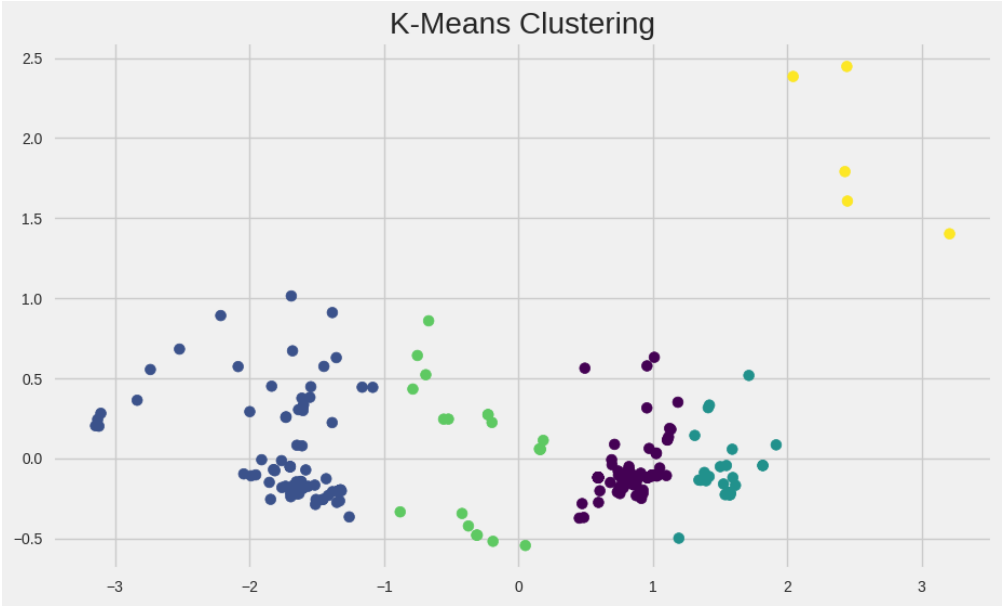

Imagen 10. Gráfica de dispersión que muestra cuando k= 5,
que es uno de los valores sugeridos en el análisis de Sihlouette.

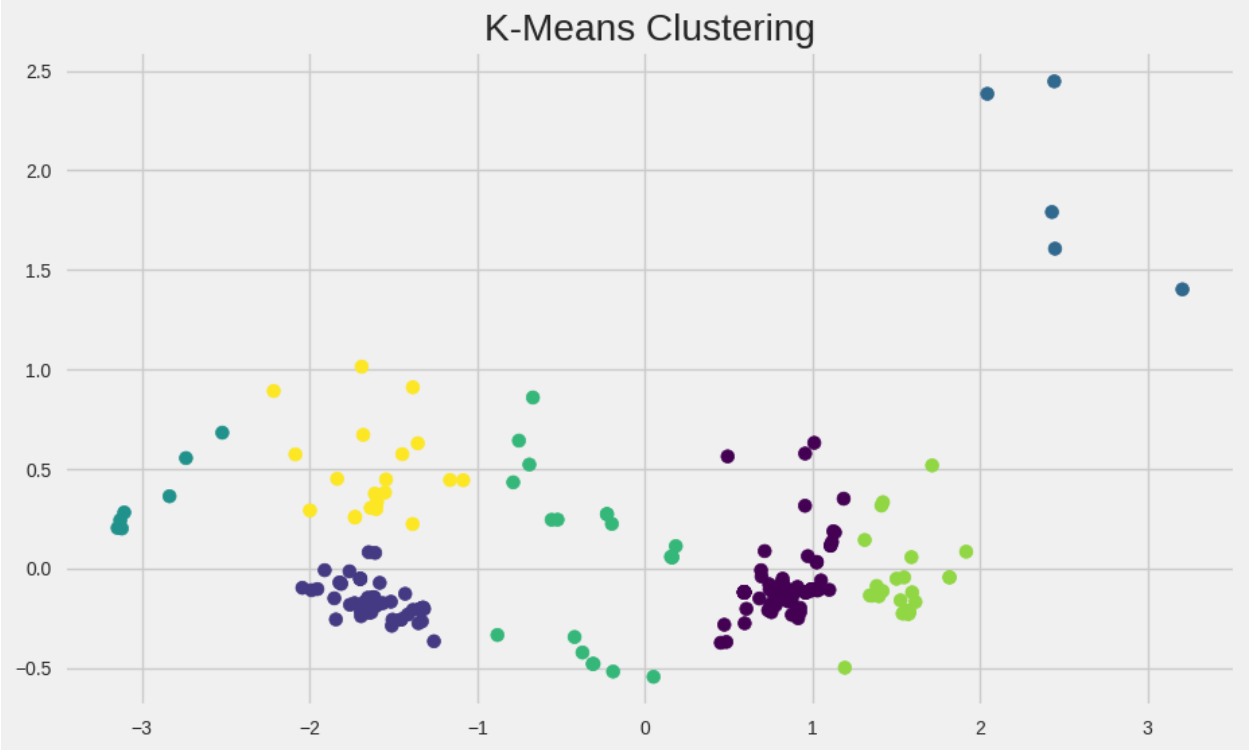

Imagen 11. Gráfica de dispersión que muestra cuando k= 7, que es uno de los valores sugeridos en el análisis de Sihlouette.

- Uso de Algoritmo Elkan: Usé el algoritmo "elkan" en lugar del algoritmo "lloyd" para mejorar la eficiencia computacional, ya que los datos que estoy usando tienen clusters bien definidos.

```
Número de iteraciones que le tomó al modelo converger a una solución
durante el ajuste.
   7
```

Imagen 12.Captura de pantalla que muestra el número de iteraciones que le toma converger cuando usa el algoritmo lloyd.

```
Número de iteraciones que le tomó al modelo converger a una solución
durante el ajuste.
   3
```

Imagen 13.Captura de pantalla que muestra el número de iteraciones que le toma converger cuando usa el algoritmo elkan.

- Uso de k-means++: Usé el método de inicialización "k-means++" para seleccionar los centroides iniciales. Esto mejora la convergencia del algoritmo y evita mínimos locales subóptimos.

Resultados y Conclusiones

En resumen, el modelo de K-Means se ha evaluado y ajustado correctamente utilizando los conjuntos de entrenamiento, prueba y validación. Se ha logrado un buen equilibrio entre bias y varianza, con un valor óptimo de k igual a 7. Además, se han aplicado técnicas de inicialización y algoritmos eficientes para mejorar el rendimiento del modelo.