Aprendizaje No Supervisado

Analítica de Datos, Universidad de San Andrés

Introducción al Aprendizaje No Supervisado

- Rama del machine learning enfocada en encontrar patrones en datos no etiquetados

Introducción al Aprendizaje No Supervisado

- Rama del machine learning enfocada en encontrar patrones en datos no etiquetados
- A diferencia del aprendizaje supervisado, no hay variable objetivo

Introducción al Aprendizaje No Supervisado

- Rama del machine learning enfocada en encontrar patrones en datos no etiquetados
- A diferencia del aprendizaje supervisado, no hay variable objetivo
- Principales objetivos:
  - Descubrir grupos naturales (clustering)
  - Reducir la dimensionalidad
  - Detectar anomalías
  - Encontrar reglas de asociación

Clustering

- Técnica que agrupa observaciones similares en clusters

Clustering

- Técnica que agrupa observaciones similares en clusters
- Los puntos parecidos se agrupan por características similares

Clustering

- Técnica que agrupa observaciones similares en clusters
- Los puntos parecidos se agrupan por características similares
- Aplicaciones:
  - Agrupar clientes según comportamiento
  - Agrupar documentos según contenido

K-Means

¿Cómo funciona?

- Se especifica el número K de clusters deseados

K-Means

¿Cómo funciona?

- Se especifica el número K de clusters deseados
- Inicialización de K centroides aleatorios

K-Means

¿Cómo funciona?

- Se especifica el número K de clusters deseados
- Inicialización de K centroides aleatorios
- Proceso iterativo:
  - Asignar cada punto al centroide más cercano
  - Recalcular centroides como promedio de puntos asignados

K-Means

¿Cómo funciona?

- Se especifica el número K de clusters deseados
- Inicialización de K centroides aleatorios
- Proceso iterativo:
  - Asignar cada punto al centroide más cercano
  - Recalcular centroides como promedio de puntos asignados
- Continúa hasta la convergencia

Implementación de K-Means

```python
from sklearn.cluster import KMeans

# Crear y entrenar el modelo
kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(X)

# Obtener centroides
centroids = kmeans.cluster_centers_
```

Reducción de Dimensionalidad

- Útil cuando tenemos datasets con muchas variables

Reducción de Dimensionalidad

- Útil cuando tenemos datasets con muchas variables
- Ayuda a:
  - Simplificar el análisis
  - Mejorar la visualización
  - Facilitar el entrenamiento de modelos

Reducción de Dimensionalidad

- Útil cuando tenemos datasets con muchas variables
- Ayuda a:
  - Simplificar el análisis
  - Mejorar la visualización
  - Facilitar el entrenamiento de modelos
- Mantiene la información más importante

PCA (Principal Component Analysis)

¿Qué hace PCA?

- Combina variables de manera especial

PCA (Principal Component Analysis)

¿Qué hace PCA?

- Combina variables de manera especial
- Crea componentes principales que:
  - Capturan la mayor variación posible
  - Son independientes entre sí

PCA (Principal Component Analysis)

¿Qué hace PCA?

- Combina variables de manera especial
- Crea componentes principales que:
  - Capturan la mayor variación posible
  - Son independientes entre sí
- El primer componente captura la mayor variación

PCA (Principal Component Analysis)

¿Qué hace PCA?

- Combina variables de manera especial
- Crea componentes principales que:
  - Capturan la mayor variación posible
  - Son independientes entre sí
- El primer componente captura la mayor variación
- Estándar: conservar 95

Implementación de PCA

```python
from sklearn.decomposition import PCA

# Crear y aplicar PCA
pca = PCA(n_components=2)  # Reducimos a 2 dimensiones
X_reduced = pca.fit_transform(X)

# Ver cuanta varianza explica cada componente
expl_variance = pca.explained_variance_ratio_
print(f"var. explicada por cada componente: {expl_variance}")
print(f"var. total explicada: {sum(expl_variance):.2f}")
```

¿Cuándo usar PCA?

- **Visualización:** Para ver patrones en datos con muchas variables

¿Cuándo usar PCA?

- **Visualización:** Para ver patrones en datos con muchas variables
- **Ruido:** Limpiar datos eliminando variaciones no importantes

¿Cuándo usar PCA?

- **Visualización:** Para ver patrones en datos con muchas variables
- **Ruido:** Limpiar datos eliminando variaciones no importantes
- **Colinealidad:** Con variables muy correlacionadas

¿Cuándo usar PCA?

- **Visualización:** Para ver patrones en datos con muchas variables
- **Ruido:** Limpiar datos eliminando variaciones no importantes
- **Colinealidad:** Con variables muy correlacionadas
- **Curse of Dimensionality:** Evitar problemas con muchas variables

Evaluación de Modelos No Supervisados

Clustering:

- Silhouette Score

Reducción de Dimensionalidad:

Evaluación de Modelos No Supervisados

Clustering:

- Silhouette Score
- Calinski-Harabasz Index

Reducción de Dimensionalidad:

Evaluación de Modelos No Supervisados

Clustering:

- Silhouette Score
- Calinski-Harabasz Index
- Davies-Bouldin Index

Reducción de Dimensionalidad:

Evaluación de Modelos No Supervisados

Clustering:

- Silhouette Score
- Calinski-Harabasz Index
- Davies-Bouldin Index

Reducción de Dimensionalidad:

- Varianza explicada

Evaluación de Modelos No Supervisados

Clustering:

- Silhouette Score
- Calinski-Harabasz Index
- Davies-Bouldin Index

Reducción de Dimensionalidad:

- Varianza explicada
- Reconstrucción del error

Evaluación de Modelos No Supervisados

Clustering:

- Silhouette Score
- Calinski-Harabasz Index
- Davies-Bouldin Index

Reducción de Dimensionalidad:

- Varianza explicada
- Reconstrucción del error
- Preservación de distancias

Evaluación de Clustering

```
from sklearn.metrics import silhouette_score, calinski_harabasz_score

# Evaluar clustering
silhouette_avg = silhouette_score(X, clusters)
calinski_score = calinski_harabasz_score(X, clusters)
```