Certainly! Below is a **simple, high-quality explanation** of each concept along with **only the relevant code snippets** from your script, organized by topic:

**1. DBSCAN (Density-Based Spatial Clustering)**

**Explanation:**  
DBSCAN groups data points into clusters based on density. It requires two parameters:

* eps: The radius around a point to search for neighbors.
* min\_samples: Minimum points required in the eps radius to form a dense cluster.

Points not fitting into any cluster are labeled as noise (-1).

**Relevant code:**

from sklearn.cluster import DBSCAN

dbscan = DBSCAN(eps=0.2, min\_samples=2)

clusters = dbscan.fit\_predict(scaled\_data)

data['Cluster'] = clusters

**2. Silhouette Score**

**Explanation:**  
Silhouette score measures how well each point fits within its cluster compared to other clusters.

* Ranges from -1 to 1
* Higher score means better, more distinct clusters.

**Relevant code:**

from sklearn.metrics import silhouette\_score

n\_clusters = len(set(clusters)) - (1 if -1 in set(clusters) else 0)

if n\_clusters > 1:

mask = clusters != -1 # Exclude noise points

score = silhouette\_score(scaled\_data[mask], clusters[mask])

print(f"\nSilhouette Score: {score:.40f}")

else:

print("\nSilhouette Score cannot be calculated (less than 2 clusters or all noise).")

**3. Davies–Bouldin Index (DBI)**

**Explanation:**  
DBI evaluates clustering quality by measuring the average similarity between each cluster and its most similar cluster.

* Lower DBI values are better, indicating well-separated and compact clusters.

**Relevant code:**

from sklearn.metrics import davies\_bouldin\_score

if n\_clusters > 1:

dbi\_score = davies\_bouldin\_score(scaled\_data[mask], clusters[mask])

print(f"\nDavies–Bouldin Index (DBI): {dbi\_score:.6f} (lower is better)")

else:

print("\nDBI cannot be calculated (less than 2 clusters or all noise).")

**4. Calinski–Harabasz Index (CHI)**

**Explanation:**  
CHI measures the ratio of between-cluster variance to within-cluster variance.

* Higher CHI values indicate dense and well-separated clusters.

**Relevant code:**

from sklearn.metrics import calinski\_harabasz\_score

if n\_clusters > 1:

chi\_score = calinski\_harabasz\_score(scaled\_data[mask], clusters[mask])

print(f"Calinski–Harabasz Index (CHI): {chi\_score:.6f} (higher is better)")

else:

print("\nCHI cannot be calculated (less than 2 clusters or all noise).")

**5. ANOVA Test for Statistical Significance Between Clusters**

**Explanation:**  
ANOVA tests if the means of features (like Satisfaction and Loyalty) differ significantly across clusters.

**Relevant code:**

from scipy.stats import f\_oneway

clustered\_data = data[data['Cluster'] != -1]

satisfaction\_groups = [group['Satisfaction'].values for \_, group in clustered\_data.groupby('Cluster')]

loyalty\_groups = [group['Loyalty'].values for \_, group in clustered\_data.groupby('Cluster')]

f\_stat\_satis, p\_val\_satis = f\_oneway(\*satisfaction\_groups)

print(f"\nANOVA P-value (Satisfaction across clusters): {p\_val\_satis:.6f}")

f\_stat\_loyal, p\_val\_loyal = f\_oneway(\*loyalty\_groups)

print(f"ANOVA P-value (Loyalty across clusters): {p\_val\_loyal:.6f}")

If you want, I can help you turn these into well-commented functions or integrate explanations into a full report!