### **1. Algorithm Overview (20%)**

### 

### 

### 

### 

### 

### **2. Algorithm Comparison (40%)**

### 

### **3. Table Update (20%)**

**Compare and contrast characteristics for all three algorithms:**

| **Feature** | **k-Means** | **Hierarchical Clustering** | **DBSCAN** |
| --- | --- | --- | --- |
| **Definition** | Partitioning algorithm that assigns points to k clusters based on centroids | Builds a hierarchy of clusters using distance metrics |  |
| **Approach** | Iteratively minimizes variance within k clusters | Agglomerative (bottom-up) or divisive (top-down) |  |
| **Number of Clusters** | Requires predefined k | Can be determined from dendrogram but subjective |  |
| **Cluster Shape** | Prefers spherical clusters | Works well with various shapes but can be unstable |  |
| **Initialization** | Randomly selects k initial centroids | No initialization needed |  |
| **Result** | Hard assignments—each point belongs to a single cluster | Hierarchical structure (tree/dendrogram) |  |
| **Interpretability** | Moderate—cluster assignments but no hierarchy | High—dendrogram can be analyzed |  |
| **Strengths** | Simple, fast and efficient on large datasets | Can capture hierarchical relationships |  |
| **Limitations** | Sensitive to initial centroids and k choice | Computationally expensive for large datasets |  |

### 

### 

### **4. Code Documentation & Submission Quality (20%)**

<link to GitHub repository / code here>