**Satellite Clustering and Anomaly Detection – Solution by Nisrin Dhoondia  
  
TLE Data → sgp4 Feature Extraction → Scaling → DBSCAN Clustering → Orbit Type Mapping (LEO/MEO/GEO) → IsolationForest Anomaly Detection → Outputs (Clusters + Anomalies + Rates + Plots/CSVs).**

**1. Approach Overview**

We worked on a pipeline that takes **satellite TLE (Two-Line Element) data** and extracts **real orbital mechanics parameters** using the **sgp4 library**.  
We selected a few necessary meaningful orbital parameters that actually matters instead of using all raw features. Also, I did a bit of feature engineering — for example, converting no\_kozai into **revolutions per day**, which makes it easier to map satellites into LEO/MEO/GEO.

* Inclination (orbit tilt)
* Eccentricity (shape of orbit)
* **Mean motion** (how many orbits per day)

This gives us a **scientifically grounded feature space** where clusters naturally map to known orbit types.

**2. Clustering (DBSCAN)**

* **We used DBSCAN on the scaled features.**
* DBSCAN is useful here because it doesn’t need the number of clusters in advance and can handle noise.
* DBSCAN Performance Metrics:
  + **Silhouette Score**: 0.835 (good separation, close to 1)
  + **Davies-Bouldin Index**: 0.140 (very low, good compactness/separation)
  + **Calinski-Harabasz Index**: 10738 (high, strong clustering quality)
* Cluster Counts:
  + Cluster 0 → 8073 satellites
  + Cluster 1 → 147 satellites
  + Cluster 2 → 258 satellites
  + Cluster 3 → 64 satellites
  + Noise (-1) → 90 satellites

These metrics confirm that DBSCAN created **well-formed clusters** matching orbital regimes.

* Output Files:
  + DBSCAN\_Clustering\_Metrics.csv → clustering metrics + counts
  + DBSCAN\_Clustering.png → scatterplot of clusters (inclination vs mean motion by cluster) -> shows how satellites group into orbital “highways.”
* **After clustering, we looked at the mean motion values in each group to assign them to:**
  + **LEO (Low Earth Orbit)**: >12 rev/day (fast, close to Earth)
  + **MEO (Medium Earth Orbit)**: 2–12 rev/day (medium altitude, e.g., GPS)
  + **GEO (Geostationary Orbit)**: ~1 rev/day (stationary above Earth)
* Result:
* **Clusters line up well with these orbit regimes.**
* DBSCAN also marks some objects as **noise**, which often corresponds to unusual or rare satellites.
* **We created a mapping:**
* **LEO clusters** → dense groups, many satellites and debris
* **MEO clusters** → medium-altitude navigation satellites
* **GEO** → very sparse (DBSCAN can mislabel sparse GEO objects as noise)
* **Noise** → scattered objects that don’t fit neatly (possible anomalies)
* Output Files:
  + Cluster\_Interpretations.csv → orbit type + comments for each cluster
* SatelliteClusters\_Before\_AnomalyDetection.csv → clusters before anomaly filtering
* DBSCAN\_Clustering\_OrbitType.png → clusters labelled as LEO/MEO/GEO

**3. Anomaly Detection (IsolationForest)**

* **Inside each non-noise cluster, we applied IsolationForest to flag unusual satellites:**
* Marks objects as **normal (1)** or **anomaly (-1)**.
* Anomalies are satellites that don’t behave like the rest of their cluster.
* **Anomalies Found - the inference is based on combining the object type from Object name from Dataset with only the anomaly output from IsolationForest:**
* **Debris** drifting away from its group
* **Rocket bodies (R/B)** possibly tumbling or decaying
* **Active satellites** with motions not matching their cluster neighbors
* Output File:
* Flagged\_Anomalies.csv → detailed anomalies with explanations
* DBSCAN\_IsolationForest\_Anomalies.png → clusters with Anomalies Highlighted: red “X” marks show the odd satellites inside normal groups.
* DBSCAN\_IsolationForest\_Anomalies\_Clean.png -> clean anomalies view (noise removed) -> same anomaly plot as above plot but without DBSCAN noise, so it’s easier to focus on real clusters.
* **Anomaly Rate by Object Type:**

We classified satellites into **Debris, Rocket Bodies, Satellites** and measured anomaly rates.

* Output File:
* AnomalyRate\_by\_ObjectType.csv → anomaly counts & % per object type
* **Cluster-Specific Contamination Rate:**

Certain clusters are less stable and contain more anomalies.

* Output File:
  + Cluster\_Contamination\_Rates.csv → anomalies per cluster (% contamination)

**4. Why This Matters**

* **Interpretability:** results are not just ML outputs, but **real orbital regimes**
* **Credibility:** grounded in **orbital mechanics (sgp4 + TLEs)**, not just arbitrary features but extracted real orbital parameters
* **Value:**
  + Cluster insights → “traffic lanes” in space (LEO/MEO/GEO)
  + Anomaly insights → “problem objects” (debris, rockets, unstable satellites)
* This approach connects machine learning with real orbital science, so the clusters are not just math outputs — they’re meaningful categories which can be explained in plain language
* **Interpret anomalies based on object type (satellite, debris, rocket body) and context** → use them for **SSA tasks** like collision prevention, satellite health monitoring, and threat detection.

**In one line:**  
We clustered **satellites into LEO/MEO/GEO** using orbital mechanics, then flagged **debris, rocket bodies, and unusual satellites** as anomalies. This gives both **space traffic insights** (clusters) and **problem object alerts** (anomalies), in a data-driven yet physics-grounded way.

TLE → sgp4 → Scaling → DBSCAN → Orbit Types → IsolationForest → Outputs