**Vendor Segmentation & Risk Detection - Project Report**

**Overview**

This project groups vendors based on how much they were paid and how they performed. This project uses machine learning to group vendors based on their transaction behaviour and to find any unusual or risky vendor activity. It uses unsupervised learning, which means we don't tell the model what to look for — it finds patterns by itself.

**Project Structure**

* **data/**: Raw and processed CSVs
* **notebooks/**:
  + 01\_data\_processing.ipynb: Cleans and encodes vendor data
  + 02\_clustering.ipynb: Performs K-Means clustering
  + 03\_anomaly\_detection.ipynb: Detects anomalies using Isolation Forest
* **src/**:
  + data\_processing.py: Functions to clean and prepare data
  + clustering.py: Functions for clustering
  + anomaly.py: Functions for anomaly detection
  + utils.py: Shared utility functions (plots, save, summary)
* **04\_dashboard.py**: Streamlit dashboard for visualization

**Step 1: Data Processing**

**Goal:** Prepare the data for analysis.

**Key Tasks:**

* Remove rows with zero or negative Award\_Amount
* Create a new column: Log\_Award\_Amount = log(1 + Award\_Amount)
* Label encode categorical columns like Award\_Agency, Award\_Type, etc.

**Why log transform?**

* It compresses large values to make patterns easier to detect

**Why encoding?**

* Machine learning models work only with numbers — so categories must be converted to numeric codes

**Step 2: Vendor Clustering**

**Goal:** Group vendors into similar clusters using K-Means

**Algorithm:**

* K-Means Clustering
* Formula for distance: Euclidean
* Select k (number of clusters) using the **Elbow Method**

**What is K-Means?**

* K-Means tries to group data into k clusters based on feature similarity
* K-Means is an algorithm that tries to group data points into 'K' number of groups based on similarity.  
  - It finds cluster centres (called centroids).  
  - It assigns each vendor to the nearest centroid.  
  - It does this repeatedly until the clusters are stable.
* Formula used to calculate similarity: Euclidean Distance  
  distance = √((x2 - x1)² + (y2 - y1)²)

**What is the Elbow Method?**

* It's a way to choose the best value for k
* We run K-Means for different values of k and plot the total error (inertia)
* When the error stops decreasing quickly (the "elbow" point), that k is likely the best

**Features used for Clustering:**

* Log\_Award\_Amount, encoded versions of award type, agency, etc.

**Evaluation:**

* **Silhouette Score**:

To check how well the clustering worked:  
 - Score near 1 = good clustering  
 - Score near 0 = overlapping clusters

* **PCA**:

PCA is used to reduce the number of columns (features) for visualization.  
 - It finds new directions (called components) where the data varies the most.  
 - We reduce the data to 2 components: PCA1 and PCA2, for a 2D plot.  
This helps us visualize clusters and outliers clearly.

**Step 3: Anomaly Detection**

**Goal:** Flag vendors whose behaviour is very different from most others

**Algorithm:**

**Isolation Forest:** Isolation Forest is used to find outliers (unusual vendors). It works by creating decision trees that split the data.  
 - Vendors that get separated early in the tree are considered anomalies.  
 - It’s called 'Isolation' because it isolates the rare patterns fast.  
 - We set a contamination level (like 5%) to flag the most unusual vendors.

* **How it works**: It randomly splits the data and sees how easily a point can be isolated
* Outliers are points that get isolated quickly

**Output:**

* Label each vendor as **"Normal"** or **"Anomaly"**

**Visuals:**

* PCA 2D scatter plot with red for anomalies and blue for normal vendors

**Step 4: Dashboard**

**Tool:**

* **Streamlit:**

The Streamlit dashboard allows users to:  
- Select a vendor cluster  
- Filter by anomaly or normal  
- View spend distribution and PCA scatter plots

**Features:**

* Load processed data
* Visualize clustering and anomalies
* Easy dropdowns to explore vendor profiles

**Results Summary**

* Vendors were grouped into 4 clusters.
* Data cleaned and processed for 1000+ vendors
* Best clustering k found using Elbow and Silhouette methods
* Isolation Forest flagged ~5% vendors as anomalous
* Dashboard helps filter and explore vendor patterns.

**Future Improvements**

* Use DBSCAN or HDBSCAN for better unsupervised clustering
* Try other anomaly models (e.g., One-Class SVM)
* Connect to real-time procurement database
* Add role-based login to dashboard

**Author**

**Aakanksha Haravde**

**Pullangari Vedaprakash**

**Atharva Shailendra Patil**