**System Log Anomaly Detection**

**Introduction:**  
This application aims to detect anomalies in the dataset uploaded by users. It pre-processes message content, clusters the data, and visualizes anomalies for easier understanding. Additionally, it provides a downloadable file highlighting the detected anomalies using the Local Outlier Factor (LOF), where anomalies are marked with a value of "-1".

**Problem Statement:**

In modern computing environments, large volumes of system logs are generated daily. Identifying anomalies within these logs is crucial for detecting security threats, diagnosing system failures, and ensuring operational efficiency. However, manual analysis of logs is time-consuming, inefficient, and prone to human error. This project aims to automate the process by using machine learning techniques for log preprocessing, clustering, and anomaly detection. Since the dataset has no predefined ground truth, the approach focuses on detecting anomalies based on base words, leveraging text processing and clustering techniques. Through an interactive Streamlit-based interface, users can efficiently upload logs, visualize anomalies, and download processed results, making anomaly detection more accessible and effective.

**Tech Stack:**

* Streamlit (For UI)
* Python (Language Used)
* Libraries
  + NLTK (For tokenization, Lemmatization and stop words removal)
  + Pandas & Numpy (For data pre-processing)
  + Scikit-Learn (anomaly detection, clustering, TF-IDF)
  + Seaborn & Matplotlib (Visualizations)

**WorkFlow:**

**1.Libraries Import and Download**

* The required libraries are imported into the program.
* Certain NLTK files may not be preloaded, so they are downloaded each time the code runs.

**2.File Upload**

Streamlit provides a user-friendly interface with a file uploader component, allowing users to dynamically select files for analysis.

**3.Data Pre-processing**

Some files may have their first column dropped when uploaded via read\_csv (Pandas function). This is handled accordingly to ensure the integrity of the message column.

The ‘preprocess\_log’ function performs:

* Tokenization: Splitting text into individual words.
* Stop-word removal: Eliminating common words that do not add value.
* Lemmatization: Converting words to their root form.

The message/Message column is extracted and processed.

* TF-IDF:Categorical data is converted into numerical form for analysis.

**4.Determining the Number of Clusters**

Two key metrics are used to determine the optimal number of clusters:

* Silhouette Score Measures how well-separated clusters are.

Higher Silhouette Score → Better clustering.

* Davies-Bouldin Score Measures compactness & separation of clusters.

Lower DB Score → Better clustering.

In the ‘determine\_optimal\_clusters’ function :

* Sets a range for number of clusters(k)
* Uses MinibatchKMeans(for memory-efficiency) algorithm for clustering.
* The k value with optimal results for the scores → Final K value.

**5.Dimisionality Reduction**

* After K value is determined, clustering is performed.
* PCA is done to reduce the dimensionality to 2D.
* Next Step → anomaly detection

**6.Anomaly Detection**

* Local Outlier Factor (LOF) is an unsupervised anomaly detection algorithm that identifies outliers based on their distance from neighbouring points.
* The ‘n\_neighbors’parameter plays a crucial role in accuracy, determining how many nearby points are considered.
* A scatter plot visualizes clustering results, with anomalies highlighted in red.

**7. Anomaly Statistics**

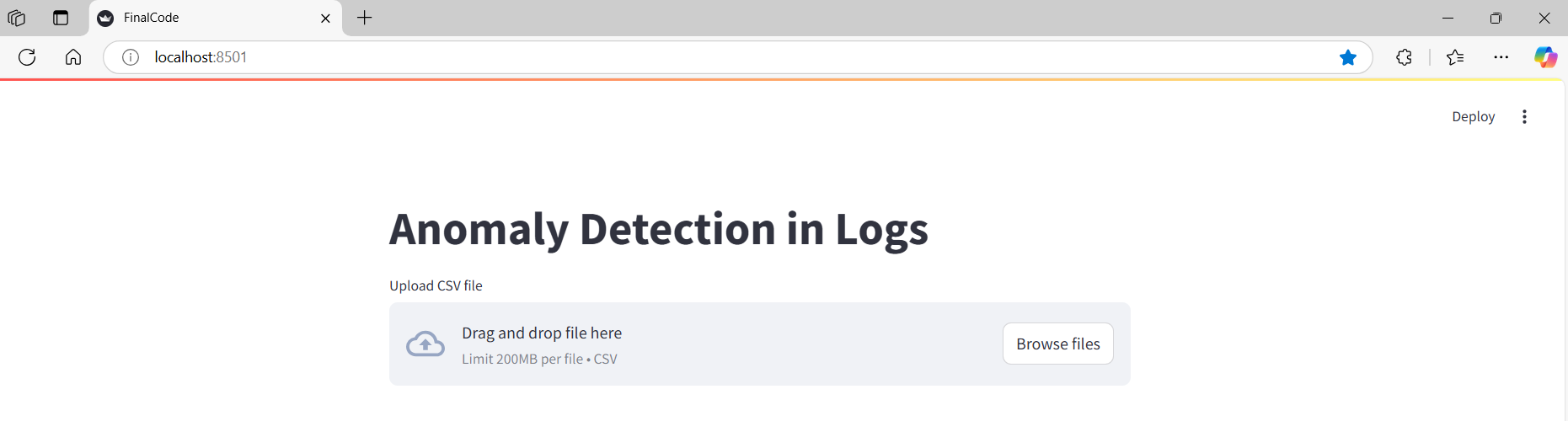
Displays key dataset statistics:

* Total number of records.
* Number of detected anomalies.
* Percentage of anomalies.
* Final Silhouette and Davies-Bouldin scores for clustering.

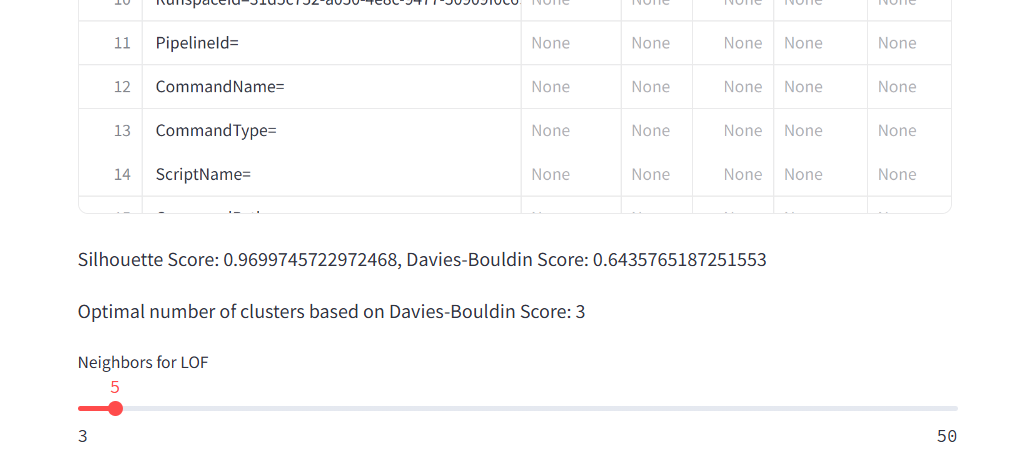
**8.Download file with LOF anomalies marked**

* The processed data (with cluster and anomaly labels) is saved as a CSV file.
* A download button is provided in Streamlit for users to obtain the processed file.

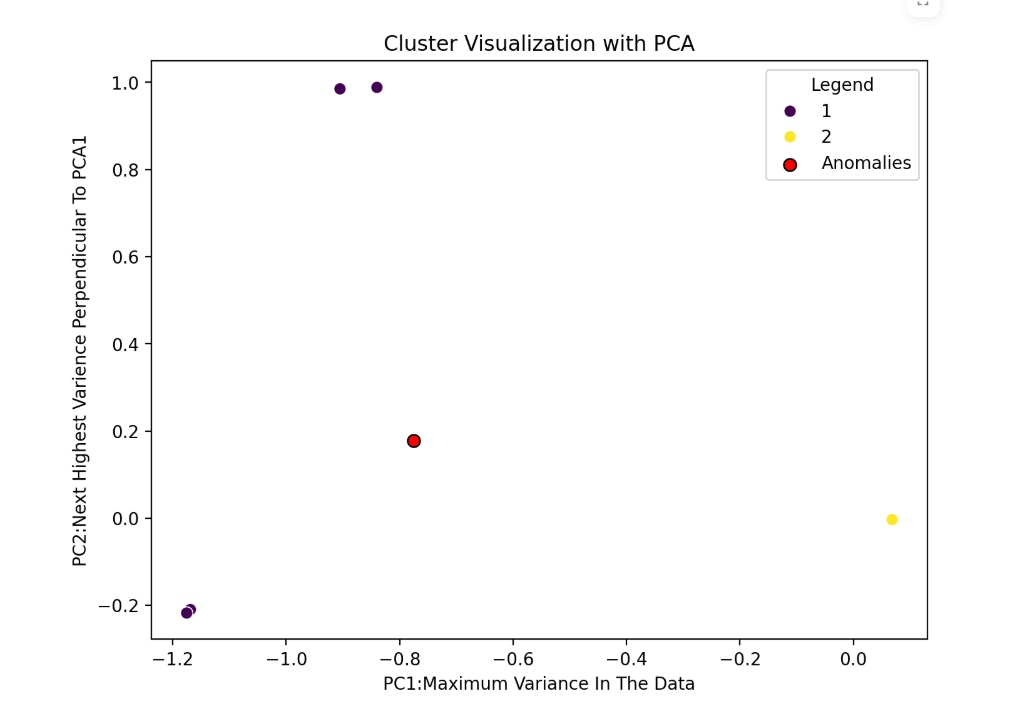
**Working:**

****

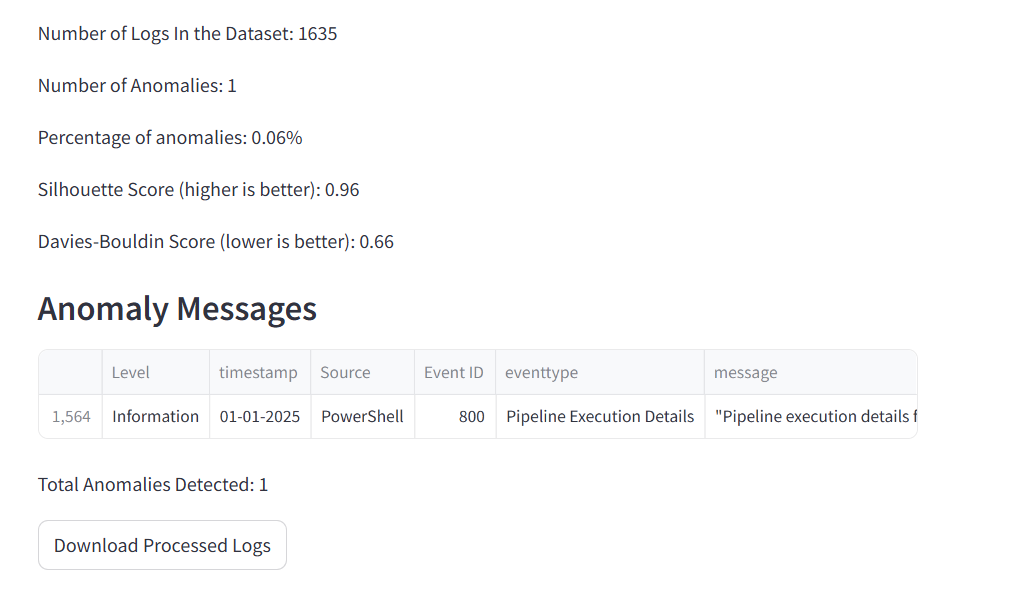
1.The Image shows File Upload Option



2. The image displays the uploaded dataset, the determined optimal number of clusters, and a slider that allows users to adjust the LOF (Local Outlier Factor) neighbors value for anomaly detection.



3.The Clusters are plotted(Anomalies –red)



4.The Anomaly statistics, the Anomaly record and the download option is shown.

**Diagram:**

