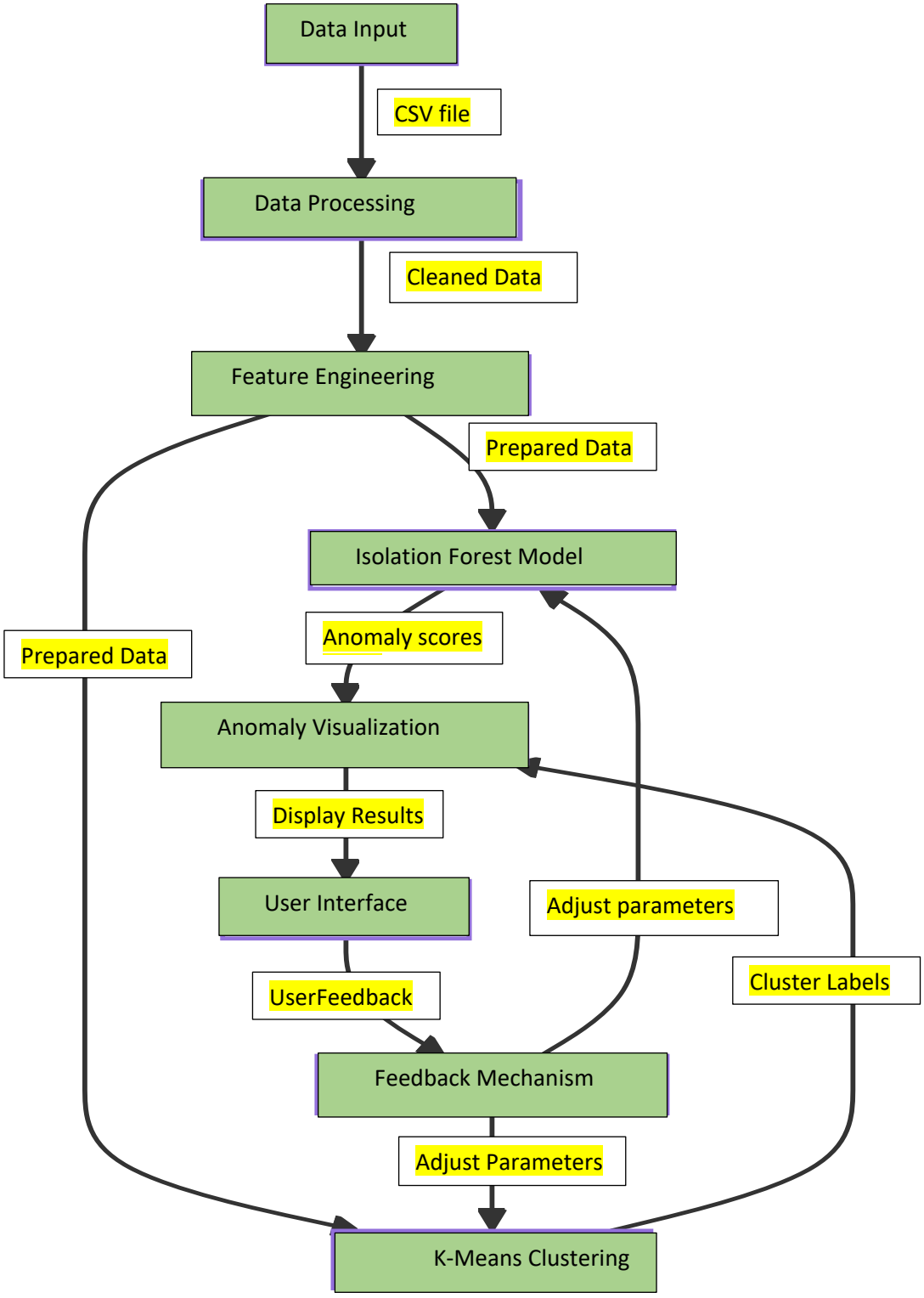
Architecture Diagram:



Explanation of the Diagram Components:

1. **Data Input**: The initial step where data is received, typically in the form of a CSV file.

2. **Data Preprocessing**: This stage involves cleaning the data, handling missing values, and normalizing it for further analysis.

3. **Feature Engineering**: Relevant features are extracted or transformed from the cleaned data to improve model performance.
4. **Anomaly Detection (Isolation Forest):**

   We use the Isolation Forest model for anomaly detection. The contamination parameter is set to 0.05, meaning we expect about 5% of the data to be anomalies. The model predicts anomalies (-1 for anomaly and 1 for normal), and we map the predictions to Normal and Anomaly labels.
   Detected anomalies are displayed in the app.
5. **K-Means Clustering:** The user selects the number of clusters (between 2 and 10) using the st.slider() widget.
   We use the K-Means clustering algorithm to cluster the data based on the selected numeric columns.
   Cluster centers are displayed in the app.
   The app then visualizes the clustering result using a scatter plot, where each point is colored based on its cluster.
   Anomalies and Clusters:
   In the scatter plot of clusters, anomalies (detected by Isolation Forest) are overlaid in red for easy visualization.

   Two plots are shown: one for clustering and another showing clustering results along with the detected anomalies.
6. **Anomaly Visualization**: Seaborn is used for creating the scatter plots, and Matplotlib is used for customizing the visuals.The first plot shows the clusters, and the second plot highlights anomalies within the clusters.

7. **Display Results**: The visualized results are presented to the user through an interface.

8. **User Feedback**: Users can provide feedback on the results, which can be used to adjust parameters in both the Isolation Forest and K-Means models for enhanced accuracy.

Notes :

Anomaly Detection: You can adjust the contamination parameter in the IsolationForest to change the proportion of expected anomalies.

Clustering: You can experiment with other clustering algorithms, like DBSCAN or Agglomerative Clustering.

Visualizations: You can enhance the visualizations by adding more customization to the plots or allowing the user to download the results.

This app gives you an interactive and easy-to-use interface for both anomaly detection and clustering in CSV data, providing insights into data patterns and outliers.