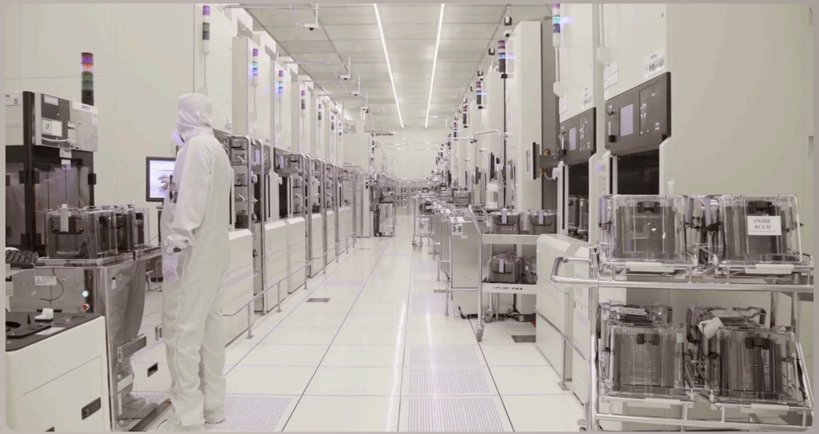
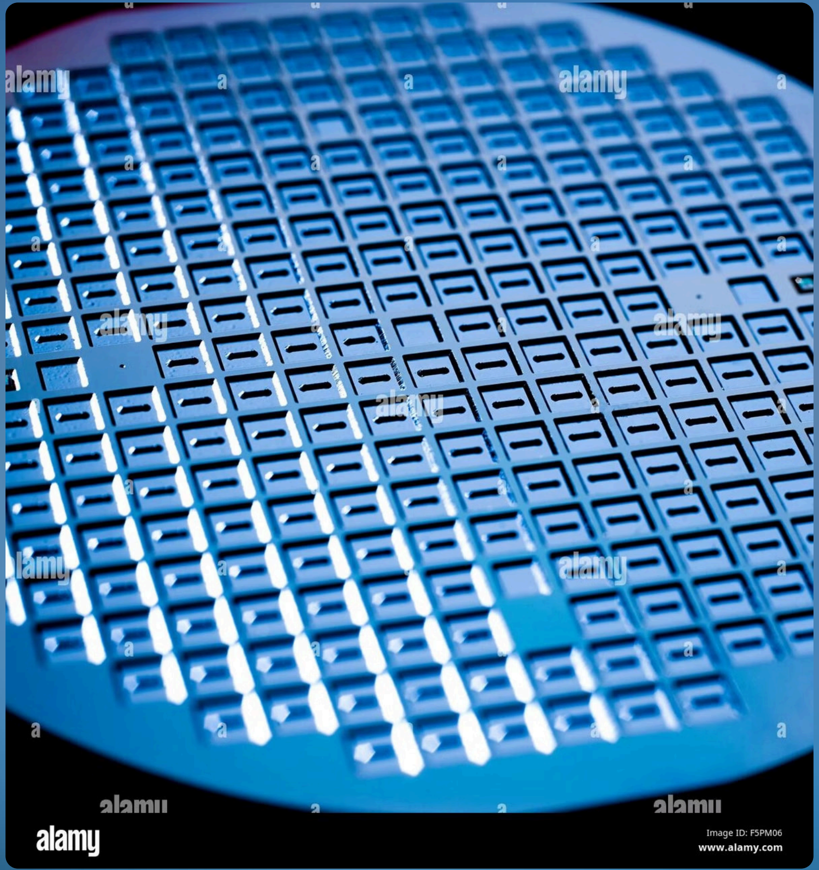
Wafer Fault Detection

Revolutionizing semiconductor manufacturing through advanced machine learning, our Wafer Fault Detection System provides real-time predictive analysis for enhanced quality control.

by Raveen Kumar





Understanding Wafer Materials

Wafer Type	Crystal Structure	Applications
Silicon	Covalent Lattice	Logic circuits, microprocessors, memory
Gallium Arsenide	Zinc-Blende	RF devices, microwave amplifiers
Silicon Carbide	Tetrahedral	Power electronics, electric vehicles

The Challenge of Traditional Inspection

Manual Inspection

Time-consuming, prone to errors, and limits production throughput.

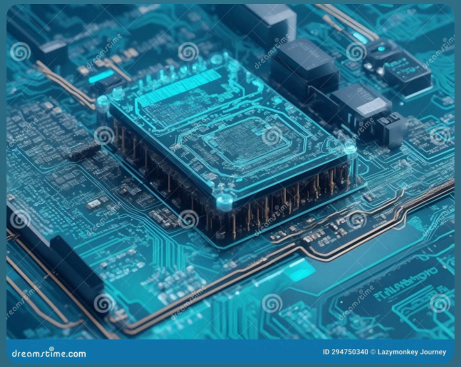
Rule-Based Systems

Limited in adaptability to complex patterns and evolving manufacturing processes.

Basic Threshold Alerts

High rate of false positives, leading to unnecessary wafer replacements.

1. **Time-Consuming**: Manual inspection is slow, especially when hundreds of wafers are involved, leading to delays.
2. **Inaccuracy**: The manual process increases the risk of false positives, where functional wafers are mistakenly flagged as faulty.
3. **Production Downtime**: Inspections often require halting production, resulting in unnecessary downtime and potential losses.
4. **Costly Errors**: Misclassifying a functional wafer as faulty can result in replacing it, leading to financial losses due to the high cost of wafers.



Machine Learning: A Smarter Approach

1 **Accurate Fault Detection**

Our robust ML model analyzes sensor data to identify faulty wafers with high accuracy.

2 **Reduced Downtime**

Minimizing manual inspection streamlines the manufacturing process and increases throughput.

3 **Data-Driven Insights**

Real-time reports and dashboards empower informed decision-making and process optimization.

Project Objectives

Accurate Fault Detection

Minimize errors in identifying faulty wafers.

Minimize Downtime

Reduce production delays for increased efficiency.

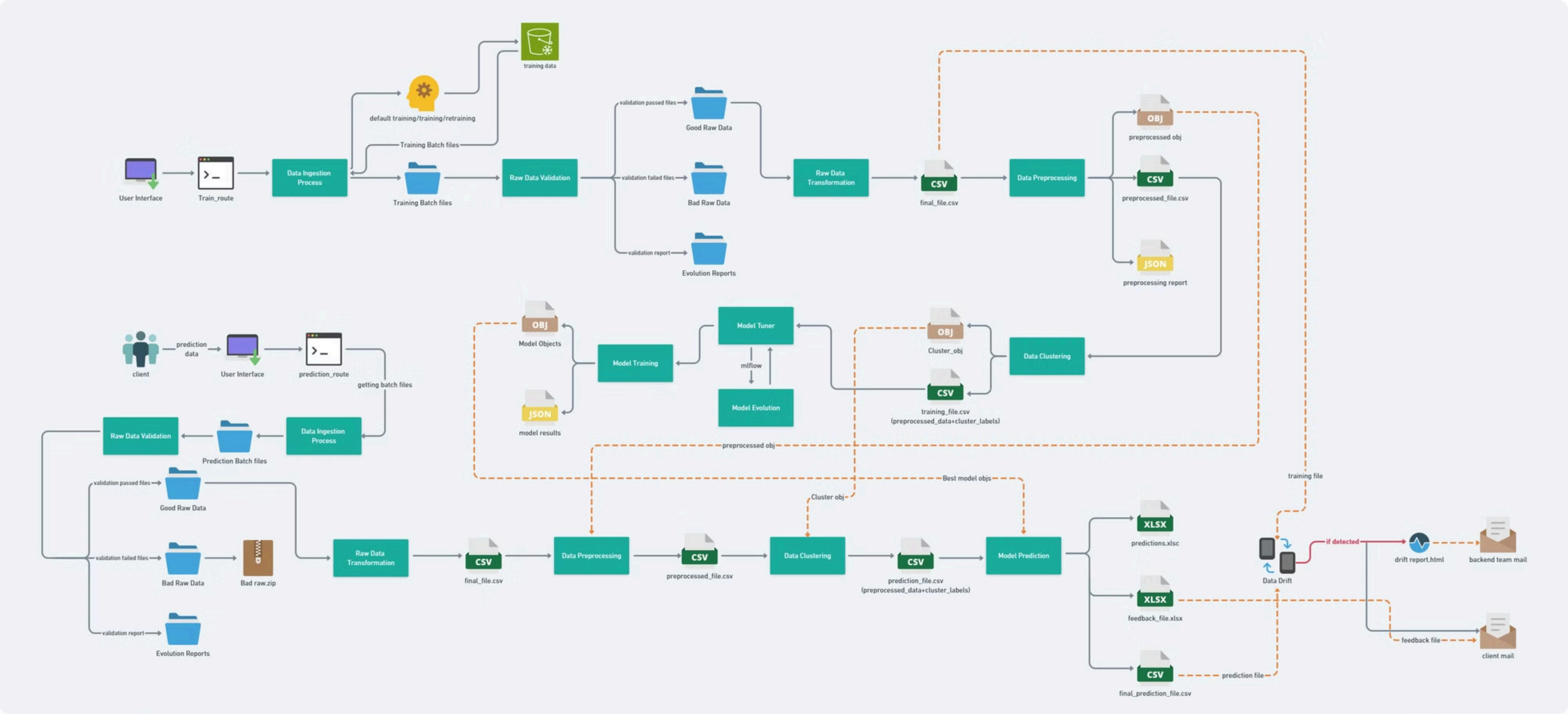
Dashboard Reports

Provide real-time insights and actionable data.

Feedback and Retraining

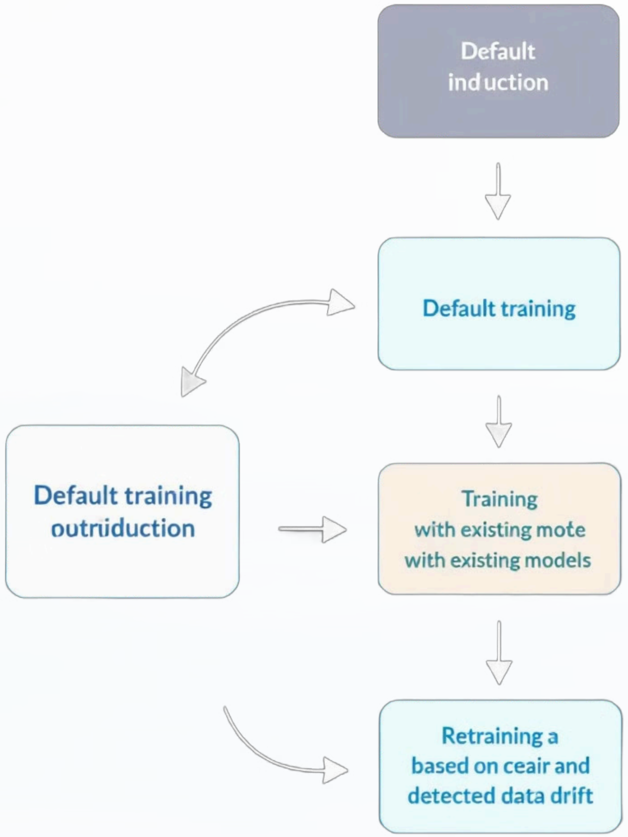
Continuously improve the model based on client feedback.

Project Overview



Data Ingestion

The system ingests data in batches, accommodating default training, training with existing models, and retraining based on detected data drift.



Data Validation

1

Bases on DSA & MDM validate the incoming files

2

File Name Validation

- validate the file name format

3

Number of Columns Validation

- Validate the number of columns

4

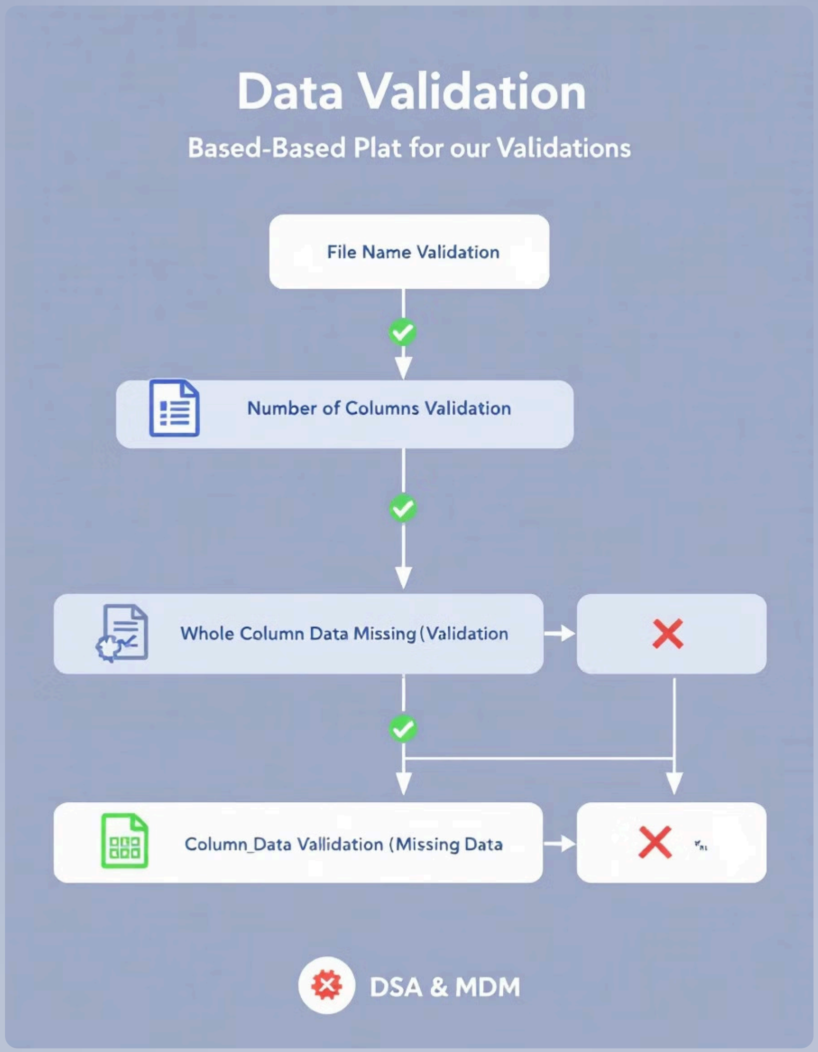
Whole Column Data Missing Validation

- Check if any column data completely missing or not

5

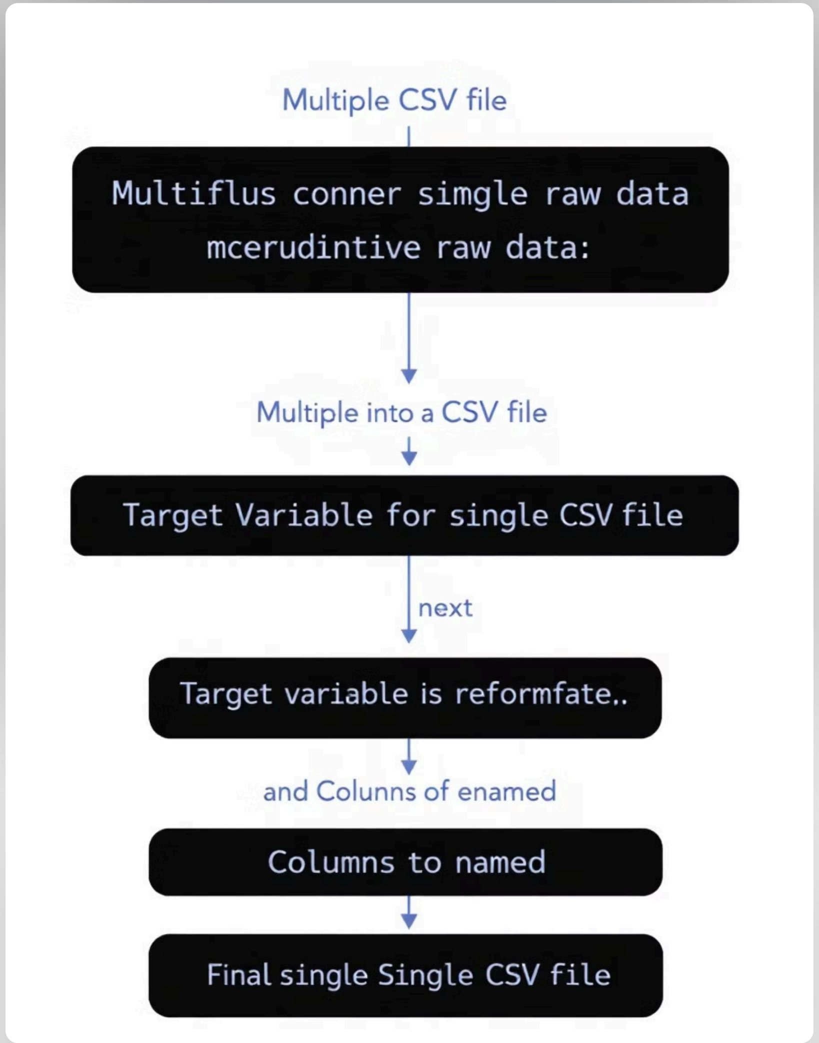
Column Data Validation

- Validates column names and data types against the schema



Data Transformation

The transformation of validated raw data by merging multiple CSV files into one, reformatting the target variable, renaming columns, and storing the result as a single file



Data Preprocessing

1

Irrelevant Column Removal

Discarding extraneous columns to focus on relevant features.

2

Duplicate Row Handling

Eliminating duplicate data points to prevent bias in the model.

3

Missing Value Imputation

This **custom transformer** handles using KNN Imputation to fill missing values, maintaining data integrity.

4

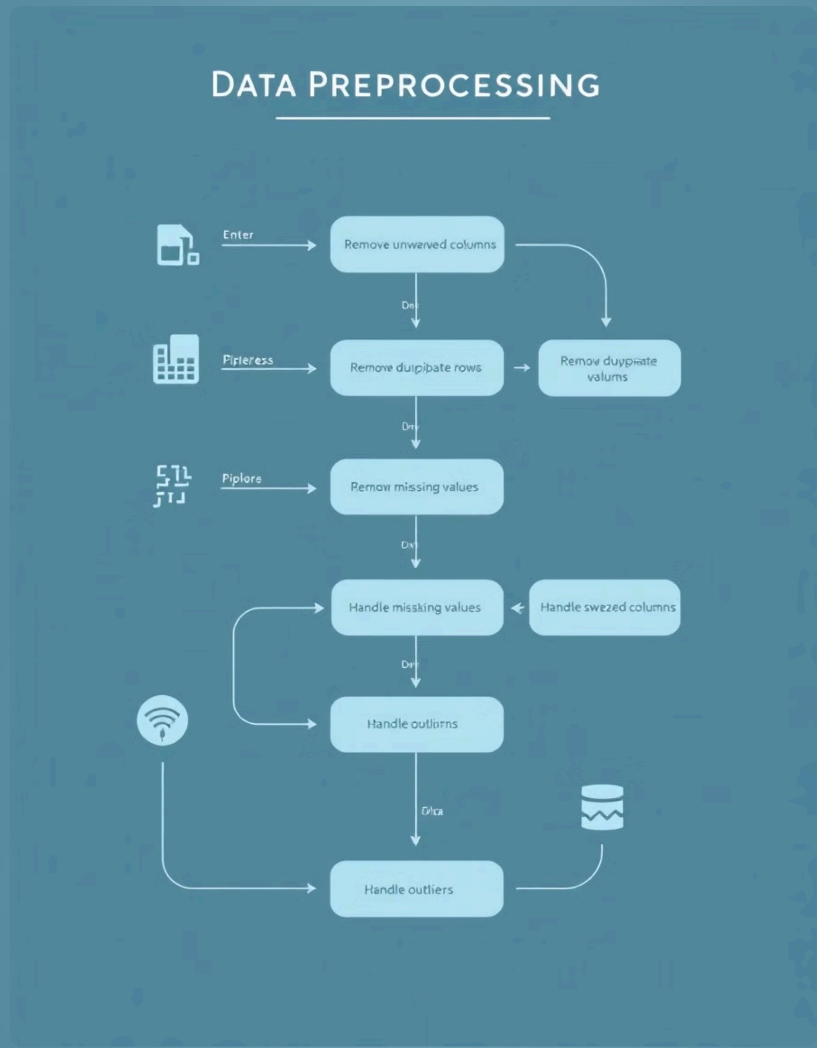
Power Transformation

This **custom transformer** handles skewness of data

5

Handle Outliers

This **custom transformer** handles outliers by clipping values based on the **IQR** method.



Data Clustering

1

Find Optimal Clusters:

- Determine the optimal number of clusters.

2

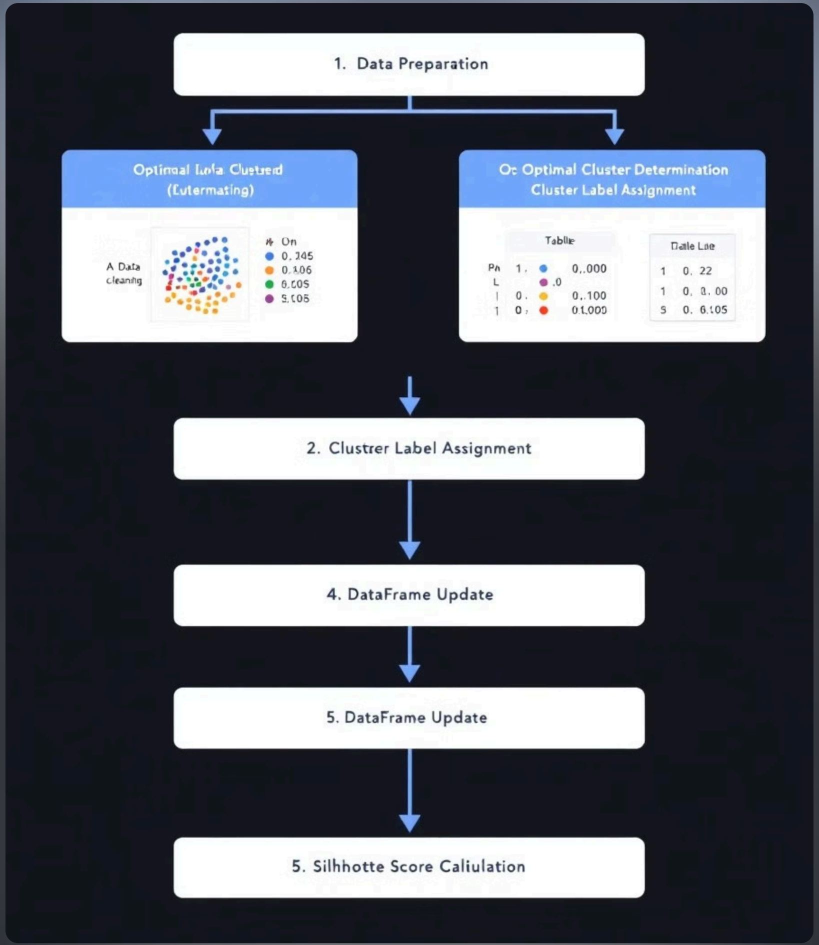
Assign Cluster Labels:

- Generate cluster labels for the data.

3

Calculate Silhouette Score:

- Creates a ClusterArtifacts object to store the final DataFrame, cluster object path, and silhouette score



Model Building

1

Experimentation with ML Models:

- Conducted initial experiments with various machine learning algorithms to identify potential models suitable for the problem.
- Models tested included standard classifiers (e.g., Logistic Regression, Decision Trees, Random Forest, XGBoost, SVM, KNN, etc.) to ensure coverage of diverse learning strategies.

2

Evaluation Criteria:

- Each model was assessed using **Cross-Validation** to ensure robust performance across different data splits.
- Performance metrics evaluated:
- **AUC (Area Under Curve)**: Measures the ability of the model to distinguish between classes.
- **Recall**: Focused on minimizing false negatives, especially important for the imbalanced dataset.
- **Precision**: Ensures that positive predictions are accurate.
- **F1-Score**: Balances precision and recall to provide a single metric.
- **Confusion Matrix Analysis**: Offers insight into true positives, true negatives, false positives, and false negatives, helping understand the model's performance visually.

3

Selection of Top 4 Models:

- Based on the evaluation scores, the top 4 models were selected.
- These models demonstrated the best combination of metrics (AUC, recall, precision, F1-score), suggesting their potential for final use.

4

Cluster-Wise Hyperparameter Tuning:

Conducted cluster-wise hyperparameter tuning for the selected models:

- Data was segmented into clusters based on similarity.
- Tailored hyperparameter tuning within each cluster to improve performance.
- Used techniques like **Grid Search** and **Random Search** to optimize parameters.

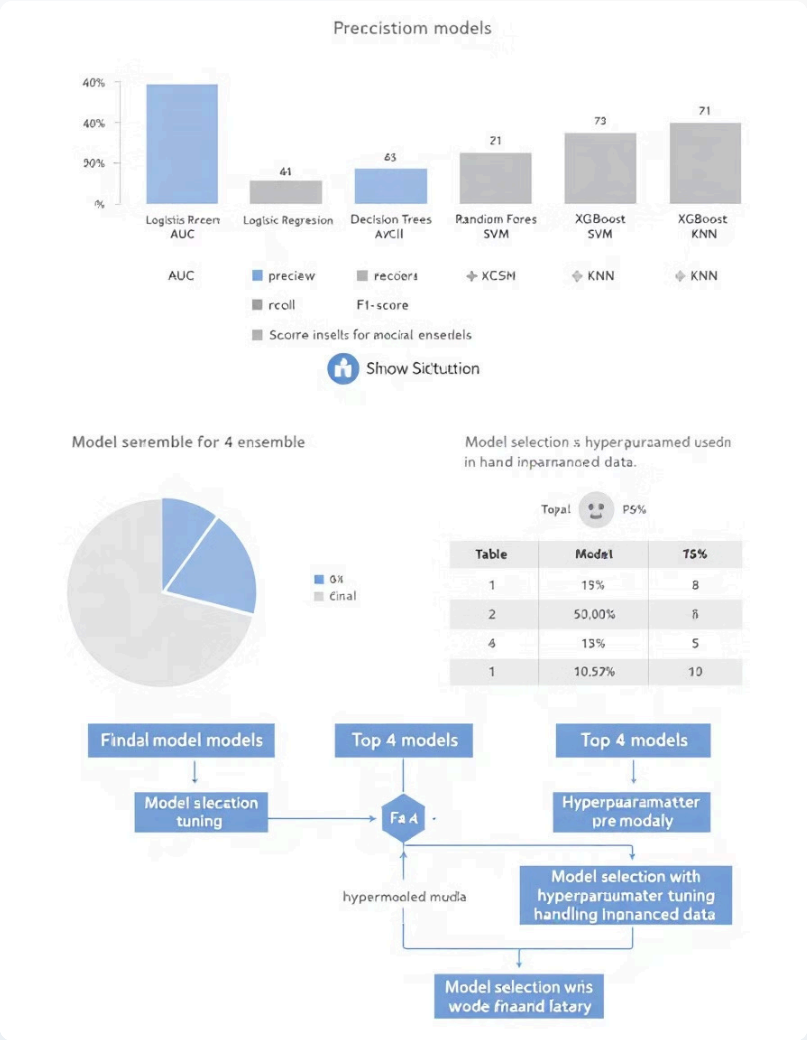
— **Handling Imbalanced Dataset**:

Implemented strategies to handle the class imbalance, including:

- **Resampling Techniques**: Such as SMOTE (Synthetic Minority Over-sampling Technique) to balance the classes.
- **Class Weights Adjustment**: Used in algorithms that support weighting to penalize misclassifications of minority classes.

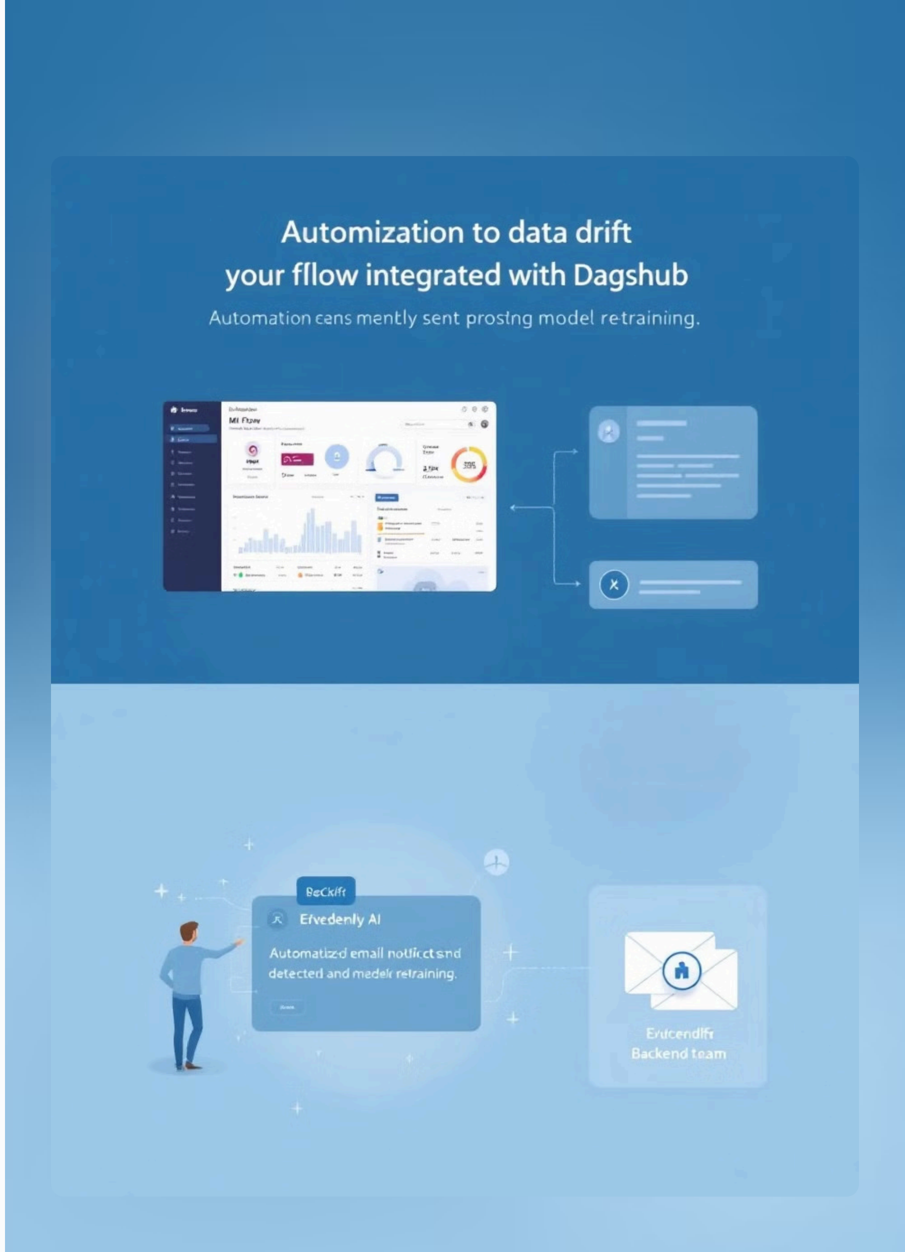
— **Dimensionality Reduction**:

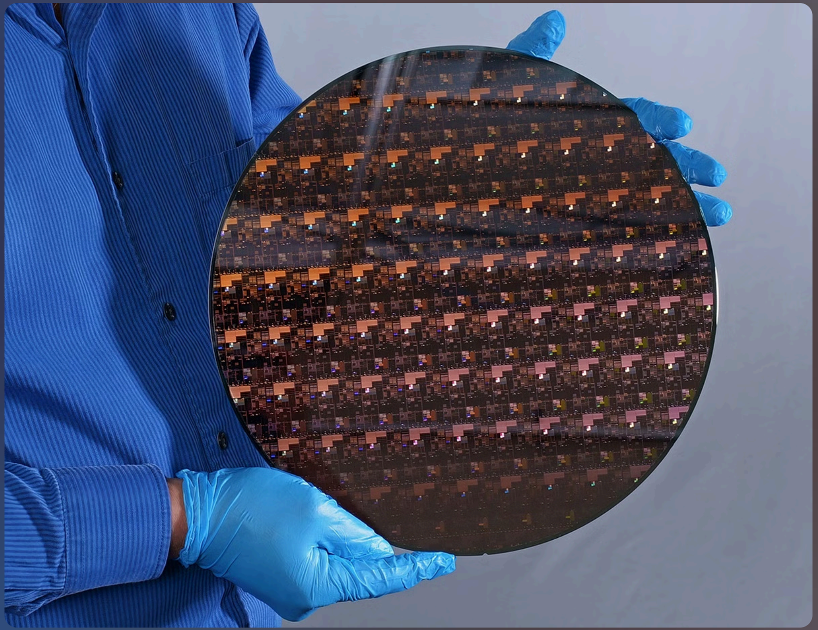
- Applied **PCA (Principal Component Analysis)** for feature decomposition to reduce dimensionality while preserving variance.
- Helped in minimizing overfitting and improving model interpretability without sacrificing significant data information.
- Selected the best-performing model(s) for deployment based on comprehensive evaluation metrics and business requirements.



MLFlow Integration and Data Drift Detection

MLflow meticulously tracks model training, performance metrics, and parameters. The system proactively detects data drift, triggering alerts and retraining to maintain model accuracy.





Benefits and Future Applications

This system significantly improves wafer quality control, reduces waste, and increases production efficiency. Future applications include integration with other manufacturing processes and predictive maintenance.