Yield Improvement in Semiconductor Manufacturing: Rare Defect Prediction with Imbalanced Learning and Nonlinear Models

Mustakim Shikalgar
Arizona State University
Tempe, AZ, United States
mshikalg@asu.edu

David Su
Arizona State University
Tempe, AZ, United States
daweisu1@asu.edu

Abstract

This study uses the SECOM semiconductor manufacturing dataset [1] to build a predictive system for rare wafer defects, in order to support yield improvement. The dataset contains hundreds of sensor features but only a very small number of failure samples. It is high dimensional, noisy, and extremely imbalanced. In this setting, if we apply standard supervised learning models directly, the model often predicts almost all wafers as "Pass", and many "Fail" wafers are missed.

We first perform data cleaning, missing value handling, and principal component analysis (PCA). The PCA plot shows that Fail samples are embedded inside a large cluster of Pass samples. This indicates that a simple linear decision boundary is not enough to separate the two classes [1, 8]. We then build several linear baseline models, including cost sensitive logistic regression and logistic regression with random oversampling. These models give a maximum Recall of around 0.2857 for the Fail class, which becomes the practical performance ceiling of linear methods in our setting.

On top of these baselines, we combine SMOTE with several nonlinear models, including Random Forest, XGBoost, and RBF kernel SVM. The results show that these models perform very well on the training set, but their Recall for the Fail class on the test set is close to 0. This is a clear sign of overfitting and model collapse. The main reason is that in high dimensional space, synthetic samples generated by SMOTE can deviate from the true data manifold, and powerful nonlinear models can overfit these artificial patterns instead of learning real defect structures.

To address this issue, we propose a two stage method that combines L1 based feature selection with tuned XGBoost.

First, we use logistic regression with L1 regularization to perform feature selection and reduce the original hundreds of sensor features to a smaller and more informative feature set. Then, on these selected features, we build a pipeline that combines SMOTE and XGBoost, and we tune hyperparameters using Randomized Search and stratified cross validation, with Recall of the Fail class as the only scoring metric. Experimental results show that the final model reaches a Recall of about 0.66 and an F1 score of about 0.30 for the Fail class on the test set. This is much better than both the linear baselines and the nonlinear models without feature selection, and it shows that our proposed pipeline can stably improve rare defect detection under imbalanced manufacturing data.

CCS Concepts

• **Computing methodologies** → **Supervised learning**; **Learning from imbalanced data**; • **Hardware** → *Semiconductor manufacturing*.

Keywords

imbalanced classification, semiconductor manufacturing, defect prediction, SMOTE, XGBoost, feature selection, LASSO, rare event detection

ACM Reference Format:
Mustakim Shikalgar and David Su. 2025. Yield Improvement in Semiconductor Manufacturing: Rare Defect Prediction with Imbalanced Learning and Nonlinear Models. In *Proceedings of December 2025 (Data Mining Project '25).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

1.1 Background and Motivation

Modern semiconductor manufacturing is highly automated and very complex. A single wafer needs to pass through hundreds of process steps, many tools, and many sensors. Small deviations in the process can cause a whole wafer to fail, which leads to significant scrap cost. Because of this, early detection of potential defective wafers is very important for yield improvement, cost control, and risk management.

In practice, fabs have collected a large amount of process and sensor data. However, the proportion of Fail wafers is usually very low. The patterns of Fail are hidden inside a huge number of normal Pass wafers. This type of problem is suitable for data mining and machine learning, but it also has several challenges:

- Very high dimensional features, strong correlations, and sensor noise
- Extremely imbalanced classes, where Fail is much rarer than Pass
- Nonlinear and overlapping decision boundaries between Pass and Fail [1]

If we directly apply standard classification models such as plain logistic regression or Random Forest without special treatment, the model often learns to predict almost all wafers as Pass. The overall accuracy can look high, but the Recall for Fail is close to zero, which is not acceptable in a real fab.

Data Mining Project '25,

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

In this context, our study focuses on the following core question: How can we build a rare defect prediction system on an extremely imbalanced semiconductor process dataset that provides stable and high Recall for the Fail class, while keeping the model reasonably stable and interpretable so that process engineers can use the results?

1.2 Problem Description and Practical Importance

The SECOM dataset used in this study comes from a semiconductor process quality inspection scenario [1]. Each record represents one wafer and its measurements after going through the process and tests. The label tells us whether the wafer passed or failed the final quality inspection, and the proportion of Fail samples is very low.

On a real production line, if the model incorrectly predicts a Fail wafer as Pass, the fab may ship risky products to customers. This can lead to complaints, returns, or even serious quality incidents. Therefore, our study gives special attention to the Recall of the Fail class. We want to reduce missed defects as much as possible, even if we must accept some extra false alarms.

If we can effectively predict wafers that are likely to fail, the fab can apply extra inspections, rework, or adjust process parameters for these high risk wafers. This can improve overall yield and lower scrap and repair costs. So this research is not only about model accuracy in a narrow sense. It is directly related to the economic and quality performance of semiconductor manufacturing.

1.3 Related Work and Existing Methods

In the research fields of imbalanced classification and process monitoring, several common methods have been studied:

- Using classic classifiers such as decision trees, Random Forest [5], SVM [6], kNN, Naive Bayes, and logistic regression, often with class weights or cost sensitive learning to handle imbalance.
- Applying oversampling methods such as SMOTE [2] and ADASYN to generate synthetic minority samples, then training standard classifiers on the balanced data.
- Using gradient boosting tree models such as XGBoost [3] with `scale_pos_weight` or class weights to adjust for class imbalance, and using feature importance to help with variable selection and model interpretation.
- For high dimensional data, combining PCA, LASSO [7], or other feature selection techniques to reduce dimensionality and lower the risk of overfitting.

However, for complex and noisy semiconductor process data, simply combining SMOTE with a strong nonlinear model is risky. Many studies have reported that such combinations can lead to very good training performance but poor test performance, especially when the minority class is very rare and the feature dimension is high. This is because SMOTE may create synthetic points that do not follow the true data structure, and powerful nonlinear models can memorize these points instead of learning robust patterns.

In our work, we observe the same phenomenon in our experiments. This motivates us to design a full pipeline that includes feature selection and regularized modeling, instead of relying on SMOTE and nonlinear models alone.

1.4 System Overview and Contributions

The machine learning system designed in this study has four main components:

(1) **Data cleaning and normalization:** We remove non numeric columns and constant features, handle missing values with median imputation, and normalize numeric features with standard scaling. This builds a stable feature space suitable for modeling.
(2) **Exploratory analysis and linear baseline models:** We use PCA to visualize the data and see that Fail samples are mostly embedded inside the Pass cluster, which suggests that a simple linear boundary is not enough. We then build several variants of logistic regression as baseline models and measure their Recall for the Fail class as the performance ceiling of linear methods.
(3) **Nonlinear models with imbalanced learning, and collapse diagnosis:** We combine SMOTE with Random Forest, XGBoost, and RBF SVM. These combinations should in theory capture complex feature interactions and make better use of the balanced training data. However, in our experiments they show very high performance on the training set but Recall near zero for Fail on the test set. This reveals serious overfitting and model collapse in this high dimensional imbalanced setting.
(4) **Final proposed method:** To solve the collapse issue, we introduce L1 based feature selection and then train XGBoost on the selected features. We use SMOTE only inside the training folds of cross validation and perform Randomized Search over key hyperparameters, using Recall as the scoring metric. This becomes our final proposed pipeline.

The main contributions of this work are:

- We demonstrate the performance ceiling of linear models and the collapse behavior of naive nonlinear models on the SECOM dataset.
- We propose an integrated pipeline that combines feature selection, imbalance handling, and regularized nonlinear modeling, and show that it significantly improves Recall for rare defects.
- We provide a reproducible and extensible machine learning pipeline for semiconductor process data mining, which can be used as a template for future studies and extended with more advanced components.

1.5 High Level Summary of Results

With linear models only, and using simple imbalance handling such as class weights and random oversampling, the Recall for the Fail class is roughly between 0.19 and 0.29. This is the practical performance ceiling of linear approaches on this dataset.

When we directly combine SMOTE with nonlinear models such as Random Forest, XGBoost, and RBF SVM, the models show excellent performance on the training data, but their test Recall for the Fail class is close to 0. This indicates strong overfitting and model collapse.

After we add L1 based feature selection and tune XGBoost on the selected features, the final model reaches a test Recall of about 0.66 and an F1 score of about 0.30 for the Fail class. This clearly

outperforms all baseline methods and nonlinear models without feature selection, and it confirms the effectiveness of our proposed method.

2 Definitions and Problem Statement

2.1 Data and Prediction Target

The SECOM dataset used in this study comes from a semiconductor process quality inspection task [1]. Each record represents one wafer and its feature values during or after the process.

- **Input features (x):** These are hundreds of continuous sensor and process variables, such as voltages, currents, temperatures, and time based measurements. After removing constant features, we still have several hundred useful features.
- **Output label (y):** The original label is (-1) and (+1). Here (-1) means Pass and (+1) means Fail. Before training the models, we convert the labels to (0, 1), where 1 means Fail (positive class) and 0 means Pass (negative class). This matches the common convention in machine learning libraries.

Our prediction target is: given the feature vector **x** for a wafer, predict whether the wafer will Fail or Pass at the final inspection.

2.2 Variables and Key Concepts

- **High dimensional sensor features:** The feature space contains many continuous variables from different tools and process steps. They can be highly correlated, redundant, and noisy, which makes modeling more difficult.
- **Class imbalance:** The number of Fail samples is much smaller than the number of Pass samples. This extreme imbalance makes standard training procedures biased toward predicting the majority class. Without special treatment, the model may reach high accuracy simply by predicting everything as Pass, while completely failing to detect Fail.
- **Recall for the Fail class:**

$$\text{Recall} = \frac{TP}{TP + FN}$$

  where higher Recall means fewer missed defects.
- **Precision and F1 Score** Precision measures how many of the wafers predicted as Fail are truly Fail. F1 is the harmonic mean of Precision and Recall. We use F1 to evaluate the trade off between detecting more defects and controlling the number of false alarms.
- **SMOTE:** SMOTE [2] creates synthetic minority samples by interpolating between existing minority samples in the feature space. It can help balance the training data, but in high dimensional space it may generate samples that are not close to the true data manifold, which can cause overfitting in nonlinear models.
- **L1 regularization:** Logistic regression with L1 regularization [7] pushes many feature coefficients to exactly zero and keeps only a small number of important features. This produces a sparse model and can be used as a feature selection method. It helps reduce dimensionality and overfitting. In this study, we use L1 as a pre step before nonlinear modeling.

2.3 Formal Problem Statement

We are given a set of training samples

$$\{(x_i, y_i)\}_{i=1}^{N}, \qquad x_i \in \mathbb{R}^d,\ y_i \in \{0, 1\}$$

where $(y_i = 1)$ means Fail and $(y_i = 0)$ means Pass, and

$$\sum_i y_i \ll N,$$

that is, Fail samples are very rare.

We want to learn a classifier

$$f : \mathbb{R}^d \to \{0, 1\}$$

that, on unseen test data, maximizes the Recall of the Fail class under reasonable Precision and F1, and that avoids severe overfitting. We want to learn a classifier $f : \mathcal{X} \to \{0, 1\}$ that maximizes the Recall of the Fail class while avoiding severe overfitting.

More concretely, we focus on the following goals:

- Among many candidate models and imbalance handling strategies, find a method that clearly improves the Recall for the Fail class compared to linear baselines.
- Build a clear and reproducible machine learning pipeline that includes data preprocessing, feature selection, sampling strategy, and model training, and that has stable performance on an independent test set.
- Provide some level of model interpretability, such as feature importance, to support future process improvement and root cause analysis by engineers.

3 Proposed Approach and System

3.1 Overall System Architecture

Our pipeline has two phases: diagnosis and optimization. The main steps are:

(1) Data cleaning and preprocessing
(2) Exploratory analysis with PCA
(3) Linear baseline modeling
(4) Nonlinear modeling with failure diagnosis
(5) L1 feature selection
(6) XGBoost training with hyperparameter tuning

The first two steps help us understand the structure of the data and build baselines. The third step shows what happens when we use nonlinear models with SMOTE [2] in a naive way. The last two steps form our proposed solution that addresses the problems observed earlier.

3.2 Baseline System

We train three variants of logistic regression:

- Plain logistic regression (no class weights)
- Cost-sensitive logistic regression (`class_weight='balanced'`)
- Random oversampling plus logistic regression

Even with these modifications, linear models reach only about 0.19 to 0.29 Recall for the Fail class, establishing the performance ceiling for linear methods on this dataset.

3.3 Nonlinear Models and Collapse Diagnosis

SMOTE combined with Random Forest [5], XGBoost [3], and RBF SVM [6] perform very well on training but have Recall close to 0 on test. This is severe overfitting due to SMOTE generating unrealistic samples in high-dimensional space. The synthetic samples deviate from the true data manifold, and powerful nonlinear models can overfit these artificial patterns instead of learning real defect structures. This finding suggests that before we apply nonlinear models with SMOTE, we should first reduce the feature space and introduce more regularization.

3.4 Proposed Method: L1 Feature Selection and Tuned XGBoost

Our final pipeline is an integrated, two-stage method designed to overcome high dimensionality, noise, and the failure of naive nonlinear models.

Stage 1: L1 Feature Selection (Noise Reduction) We implemented L1 regularization (LASSO) using LogisticRegression to enforce feature sparsity:

- `penalty='l1'` and `solver='liblinear'`
- C = **0.01** (Strong regularization to enforce sparsity)
- **class_weight =′ balanced′** (Ensures bias is accounted for during feature selection)

We then use `SelectFromModel` to choose features with non-zero coefficients. This process reduces the original 474 features to a smaller, more stable set, mitigating noise and the risk of overfitting before subsequent sampling.

Stage 2: Tuned XGBoost with SMOTE (Performance Breakthrough) The selected features are fed into an optimized nonlinear pipeline to achieve the target Recall:

- We utilize the `ImbPipeline` framework with **StratifiedKFold** to ensure SMOTE is applied only on training folds, preventing data leakage.
- We use **RandomizedSearchCV** to systematically search for optimal hyperparameters (e.g., max_depth, learning_rate) in the XGBoost model [3].
- The scoring metric is defined as Recall for the Fail class (pos_label=1), ensuring the tuning process focuses exclusively on maximizing defect detection.

This systematic approach successfully stabilized the model, achieving a final Recall of about **0.6667** for the Fail class on the test set.

4 Technical Details

4.1 Data Preprocessing

(1) **Column cleaning:** Remove non-numeric columns and zero variance features.
(2) **Label transformation:** Convert from $\{-1, +1\}$ to $\{0, 1\}$.
(3) **Missing value handling:** Median imputation using `SimpleImputer`.
(4) **Standardization:** `StandardScaler` to normalize features.
(5) **Train-test split:** Stratified 80/20 split to preserve class ratios.

4.2 Linear Baseline Methods

We implement three variants:

- **Plain Logistic Regression:** Default settings, no class weights.
- **Cost-Sensitive Logistic Regression:** `class_weight='balanced'` to increase penalty for misclassifying minority class.
- **Random Oversampling + Logistic Regression:** Duplicate minority samples before training.

4.3 Non-linear Models Combined with SMOTE

After understanding the limits of linear models, we move to nonlinear models combined with SMOTE [2], hoping to capture more complex feature relationships and improve minority class detection. We mainly consider three models:

- **SMOTE + Random forests**
  - The pipeline is: median imputation, standardization, SMOTE, then RandomForestClassifier [5].
  - Random Forest uses bagging over multiple decision trees and can model nonlinear decision boundaries.
- **SMOTE + XGBoost**
  - The pipeline is: median imputation, standardization, SMOTE, then XGBClassifier [3].
  - XGBoost is a gradient boosting tree model with regularization options and is often strong for nonlinear tabular data.
- **SMOTE + RBF kernel SVM**
  - The pipeline is: median imputation, standardization, SMOTE, then SVC (kernel='rbf', class_weight='balanced') [6].
  - RBF SVM uses a kernel mapping and margin maximization to handle nonlinear separation, and class weights help with imbalance.

In experiments, all three models perform very well on the SMOTE balanced training data. However, on the test set their Recall for the Fail class is close to 0. This shows severe overfitting to the SMOTE synthetic samples in high dimensional space. This result motivates us to add stronger feature selection and regularization before nonlinear modeling.

4.4 L1 Feature Selection Details

To reduce the impact of high dimensionality and noise, we add an L1 feature selection step before nonlinear modeling [7]. The procedure is:

- **Fit preprocessing on training data:** We apply median imputation and standardization, fitting parameters only on the training set to avoid data leakage [8].
- **Train logistic regression with L1 regularization:** We train a logistic regression model to minimize the following objective:

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \mathcal{L}(y_i, \mathbf{w}^T \mathbf{x}_i) + \lambda |\mathbf{w}|_1 \tag{1}$$

  where $\mathcal{L}$ is the logistic loss and $\lambda = 1/C$ controls regularization strength. We use a strong regularization ($C = 0.01$) to force many coefficients ($\mathbf{w}$) to exactly zero. We also use class_weight='balanced' to account for class imbalance.
- **Select features:** We use SelectFromModel to retain only the features with non-zero coefficients.

The goal of this step is to remove uninformative features, reducing the dimensionality and the risk that SMOTE will generate unrealistic samples in high-dimensional space.

4.5 XGBoost Hyperparameter Tuning on Selected Features

After L1 feature selection, we use XGBoost as our final classifier and tune its hyperparameters on the selected features.

New Pipeline Design: Since imputation and scaling are already completed in the previous step, this pipeline includes only **SMOTE** to balance the training data and **XGBClassifier** as the nonlinear classifier. This design avoids repeating preprocessing and ensures that SMOTE is applied only to training data.

Cross Validation and SMOTE Usage: We use `StratifiedKFold` with 5 folds to perform cross-validation, maintaining similar class proportions in each fold. Crucially, SMOTE is applied only on the training portion of each fold, while the validation portion retains the original distribution to prevent data leakage.

Hyperparameter Search: We use `RandomizedSearchCV` to search over the following parameter space:

- `smote__k_neighbors`: $\{3, 5, 7\}$
- `xgb_clf__n_estimators`: $\{50, 100, 200\}$ (number of trees)
- `xgb_clf__max_depth`: $\{3, 5, 7\}$(max depth of each tree)
- `xgb_clf__learning_rate`: $\{0.01, 0.1, 0.2\}$
- `xgb_clf__subsample`: $\{0.6, 0.8, 1.0\}$ (row subsampling rate)
- `xgb_clf__colsample_bytree`: $\{0.6, 0.8, 1.0\}$ (column subsampling rate)

The scoring metric is **Recall for the Fail class**, ensuring the search focuses on improving the detection rate of the minority class.

Final Model and Evaluation: After Randomized Search, we retrain the best parameter combination on the full training data. We then evaluate the model on the independent test set, recording Recall, Precision, and F1 for the Fail class, as well as the confusion matrix.

This final integrated pipeline—L1 feature selection followed by SMOTE plus tuned XGBoost—is explicitly designed to handle high dimensionality, class imbalance, and nonlinearity simultaneously.

5 Experiments and Results

This section describes our experimental setup, evaluation metrics, and a comparative analysis of baseline methods versus our proposed solution. We also analyze the contribution of specific pipeline components.

5.1 Experimental Setup and Data Description

- **Data Split:** We split the full dataset into 80% training and 20% test using stratified sampling. All models, tuning, and comparisons are performed on this fixed split.
- **Consistency of Preprocessing:** Parameters for median imputation and standardization are learned from the training set only to prevent data leakage. The test set is transformed using these fitted parameters.
- **Imbalance Handling:** For linear baselines, we use class weights and random oversampling. For nonlinear models, we use SMOTE. In our final method, SMOTE is applied strictly within cross-validation training folds.

5.2 Evaluation Metrics

Given the high cost of missing a defective wafer compared to a false alarm, our evaluation focuses on:

- **Recall (Fail Class):** The most critical metric, measuring the proportion of actual defects successfully detected.
- **F1 Score (Fail Class):** The harmonic mean of Precision and Recall, summarizing the trade-off between detection and false alarms.
- **Confusion Matrix:** Used to analyze specific error patterns (False Negatives vs. False Positives).

5.3 Baseline Model Comparison

We evaluated three logistic regression variants to establish a performance baseline. As shown in Table 1, plain logistic regression yields the lowest Recall (0.19). Adding `class_weight='balanced'` improves Recall to 0.29. Random oversampling provides a similar range.

Table 1: Linear Baseline Model Performance on Fail Class

Model	Recall	Precision	F1
Plain Logistic Regression	0.19	0.21	0.20
Cost-Sensitive LR	0.29	0.15	0.20
Random Oversampling + LR	0.24	0.17	0.20

These results indicate that the practical performance ceiling for linear models on this dataset is a Recall of approximately 0.29.

5.4 Nonlinear Model Collapse

To capture complex features, we tested nonlinear models (Random Forest, XGBoost, SVM) combined directly with SMOTE [2]. However, as shown in Table 2, these models exhibit severe overfitting.

Table 2: SMOTE + Nonlinear Models: Train vs Test Performance

Model	Train Recall	Test Recall
SMOTE + Random Forest	1.00	0.00
SMOTE + XGBoost	1.00	0.00
SMOTE + RBF SVM	0.95	0.05

While achieving near-perfect Recall on training data, the test Recall drops to near zero. This "model collapse" occurs because, in high-dimensional space, SMOTE generates synthetic samples that deviate from the true data manifold. Powerful nonlinear models memorize these artificial patterns rather than learning robust defect structures, leading to catastrophic generalization failure [4].

5.5 Effectiveness of the Proposed Method

Table 3 presents the results of our proposed pipeline: L1 feature selection followed by a tuned XGBoost classifier trained on SMOTE-augmented data.

Table 3: Proposed Method Performance on Fail Class

Method	Recall	Precision	F1
L1 Selection + Tuned XGBoost	**0.66**	0.21	**0.30**

Our method achieves a Recall of **0.66** and an F1 score of **0.30**, significantly outperforming the linear baseline ceiling (0.29).

Component Analysis:

- **Necessity of L1 Selection:** Removing the L1 step causes the Recall to become unstable and drop, confirming that reducing dimensionality is required to prevent SMOTE-induced overfitting.
- **Impact of Tuning:** Using default XGBoost parameters yielded suboptimal results. Tuning `max_depth`, `learning_rate`, and `n_estimators` was essential to control model complexity [3].
- **Scoring Metric:** Tuning using *Recall* as the scoring metric (rather than Accuracy) ensured the optimization process prioritized the minority class.

5.6 Confusion Matrix and Interpretation

Analysis of the confusion matrix shows that approximately two-thirds of the real Fail samples are correctly predicted as Fail (True Positives). This improvement comes with a trade-off in Precision (more False Positives). However, in semiconductor manufacturing, flagging a Pass wafer for re-inspection (False Positive) is an acceptable cost compared to shipping a defective product to a customer (False Negative).

Case-level analysis suggests that specific sensor variables (e.g., temperature or voltage ranges) show consistent shifts for detected Fail cases. Future work combining XGBoost[3] feature importance with SHAP analysis could map these features back to specific process tools for root cause analysis.

5.7 Summary of Results

Our experiments demonstrate that simply stacking complex models and resampling techniques is insufficient for high-dimensional, imbalanced data. By designing a pipeline that jointly considers **L1 feature selection**, **imbalance handling**, and **regularized tuning**, we successfully broke the linear performance ceiling and avoided model collapse, validating the effectiveness of our proposed approach.

6 Conclusion

This study addresses the challenging problem of rare defect prediction in semiconductor manufacturing. We demonstrated that:

(1) Standard linear approaches achieve a maximum Recall of approximately 0.29 for the Fail class.
(2) Naive application of SMOTE with powerful nonlinear models leads to complete model collapse on test data [4].
(3) Our two-stage pipeline combining L1 feature selection with tuned XGBoost achieves Recall of approximately 0.66.

The key insight is that dimensionality reduction via L1 regularization prevents SMOTE from generating unrealistic samples that deviate from the true data manifold.

Future work could explore:

- Advanced feature selection methods (mutual information, recursive feature elimination)
- Ensemble approaches combining multiple models
- SHAP values for model interpretability
- Deep learning approaches with appropriate regularization

References

[1] L. J. C. Hulse, K. C. McCarthy, and J. R. Kubica. SECOM Data Set. UCI Machine Learning Repository, 2008. https://archive.ics.uci.edu/ml/datasets/SECOM
[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.
[3] T. Chen and C. Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, San Francisco, CA, USA, 2016. ACM.
[4] H. He and E. A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.
[5] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.
[6] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.
[7] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.
[8] J. Kim, et al. Study on Data Preprocessing for Machine Learning Based on Semiconductor Manufacturing Processes. *Processes*, 12(9):1866, 2024.

Project Code

The code for this project is available at:

- https://colab.research.google.com/drive/1JFJwuuFgmLrRGo7EdhCKfZ0nXP3crAD1?usp=sharing#scrollTo=aEliZtlaCcoz

Received 02 December 2025