PERFORMANCE PREDICTION OF MANUFACTURING PROCESS USING CLASSIFICATION TECHNIQUES

Problem Statement and Overall Strategy

- **Objective** - To predict the yield of a semiconductor manufacturing process to optimize processes and increase throughput.

- **Strategy** - Perform data cleaning followed by feature selection techniques to preserve only most relevant signals. Compare and choose among various classification techniques to find the classifier that has the best fault detection performance.

Data Pre-processing

Removing/Imputing missing values

- Removed columns where the percentage of missing values >= 50% resulting in 562 columns.
- Removed columns with zero variance (constant values) as they don't contribute in decision making and ended up with 446 columns.
- Imputing the columns with < 50% missing values with *mean(x) +rnorm(length(missing(x)))*sd(x)* to prevent inflation of significance of results.

Balancing the Classes

- The dataset suffers from Rare Event problem, where majority class (Pass) is 93.4% and minority class (Fail) occurs only 6.6% among the observations, leading to severe imbalance in prediction by the fitted models (almost all the models indicated 0% sensitivity).
- We used Synthetic Data Generation and created a balanced dataset using SMOTE by over-sampling minority class and under-sampling majority class. We did not use Random Oversampling to avoid overfitting and preserve information.

Feature Selection, Model Assessment and Selection

Feature Selection

- As all of the given signals (predictors) are not equally valuable, feature selection is performed to identify the most relevant signals and eliminate noise from further data analysis.
- Principal components (PC) are created for the cleaned dataset and the first 100 PCs that explain 83% of variance in the dataset are chosen as subsets to fit classification models.
- LASSO was used for feature selection on entire predictor space and PCA was ultimately chosen as it had better performance.

Pre-processing

- **Encoding the classes:** The class "-1" of response variable is encoded as "Pass" and "1" is encoded as "Fail"
- **Merging predictors and response:** The encoded response variables are merged to the dataset containing Principal Components.
- **Train-test split:** The merged dataset thus generated is split into training set consisting of 80% of observations and test set with 20% of observations.

Model Assessment and Selection

- Models are assessed based on classification performance measures namely Sensitivity, Specificity, Accuracy and Error Rate.
- Best model will be chosen based on high Sensitivity value as well as better trade-off between Sensitivity and Specificity.

Classification Methods Used

- Logistic regression
- LDA
- kNN
- Support Vector Machines (Linear, Polynomial and Radial kernels)
- Decision Tree (Pruned, Unpruned)
- Ensemble learning (Random Forest, Bagging, Boosting)

Results - Model Comparison

Method	Prediction accuracy (%)	Sensitivity (%) (% of Fail class)	Specificity (%) (% of Pass Class)	Error Rate (%)
kNN	75.8	33.33	78.84	24.2
Logistic regression (threshold=0.88)	69.43	23.81	72.7	30.57
Tree (Unpruned-29 terminal nodes)	67.83	42.86	69.62	32.17
LDA	75.8	23.81	79.52	24.2
Tree (Pruned-terminal nodes = 16)	76.43	33.33	79.52	23.57
RF (mtry=10,ntree=5)	93.63	19.05	98.98	6.37
Bagging(mtry=100,ntree=5)	92.3	19.05	98.29	7
SVM – Linear kernel (C=1)	74.2	33.33	77.13	25.8
SVM – Radial kernel (C=5, gamma = 0.01)	86.62	14.29	91.81	13.38
SVM-Polynomial kernel(C=10,d=3)	88.85	14.29	94.18	11.14
Boosting(threshold=0.665)	76.43	38.1	79.18	23.57
Naïve Bayes	**97.77**	**90.48**	**98.3**	**2.23**

Model Selection Strategy

- As the data consists of uneven class distribution, *Sensitivity* and *Specificity* are used as performance measures than error rate. This is because in cases where the model accuracy is near ideal, the Sensitivity (% of fail cases) could be as low as 0%.

- The model that has good trade-off between Sensitivity and Specificity is chosen as the final model as we aim to accurately predict all failures while still minimizing the False Positives for ‘Pass’ cases.

Model Selection Strategy

- As per the selection process described above Naïve Bayes has the highest Sensitivity among all models – 90.48%, followed by Decision Tree – 42.86%.

- Even though Random Forest has better Specificity – 98.98% than all the other models, it is ignored as its Sensitivity is very low – 19.05%.

- Among the listed models, Naïve Bayes holds good both in terms of Sensitivity and Specificity. Hence Naïve Bayes is chosen, as it fits the dataset better in terms of good Sensitivity-Specificity trade-off and minimal error rate of 2.23%.

Conclusion - Identifying most relevant signals

- As there are a lot of irrelevant or noise signals from various sensors that hamper the prediction efficiency of statistical models, the most relevant signals are enhanced and noise is suppressed by generating Principal Components which utilized most important signals that explain 83% of the variance in process output.

- Converting signals into Principal Components highly improves prediction accuracy but at the cost of losing Model Interpretability as we cannot know the actual signals involved in prediction from the final classification model.

- As the goal of this model design is to reduce production cost by predicting the ‘Pass’ and ‘Fail’ cases in advance, we decided to proceed with PCA (as Model Interpretability doesn’t play a vital role in reducing production costs).

Conclusion

- **Increase process throughput:** Throughput in manufacturing is defined as the average output over a period of time. By using the selected model, products that would fall under ‘Pass’ category are identified, and manufacturing times could be prioritized and efficient scheduling can be done to maximize throughput.

- **Predict yield:** The yield/performance of a given process could be quantified by estimating the number of semiconductors that fall under ‘Pass’ and ‘Fail’ categories. High value for ‘Pass’ indicates that the process is efficient and provides high yield. kNN model being selected is known for its consistent performance, as it doesn’t make any assumptions about the data (non-parametric). Even when the dataset grows by a lot over time, kNN can be expected to predict yield accurately and consistently.

Conclusion

- **Reduced time to learn:** High dimensional problems have the challenge of methods being unable to converge, thereby taking a very long time to train. Using PCA for variable reduction and normalizing the data greatly reduced the dimensionality. This highly reduced the learning time of the classification models.

- **Decrease in per unit production cost:** The defective semiconductors (‘Fail’ classes) identified by the model will undergo corrective processing steps in the later stages of the manufacturing process, which allows engineers to adjust and fix them in time, thereby optimizing production and reducing costs.

Insights Gained

Principal Components

- Prediction accuracy is more for PCA model than that of LASSO.
- Data interpretability is low.

Rare Events

- Up-sampling causes overfitting, down - sampling removes valuable information.
- Hence, combination of up-sampling and down-sampling techniques (SMOTE) should be used to balance the classes.

Accuracy

- Error rate can be deceptive in rare event problems. Hence, it isn't a reliable performance measure.
- In case of Rare Events or imbalanced classes Sensitivity/Specificity are better indicators.

Seed

- The way the train-test data are split (seed used) plays a major role in model performance.