Machine Learning on Highly Imbalanced Manufacturing Data Set

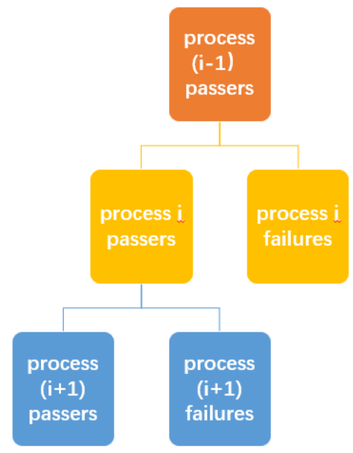
Liyu Ma, Prof. R. Collado

STEVENS INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

Statistical Learning & Analytics
Fall, 2017

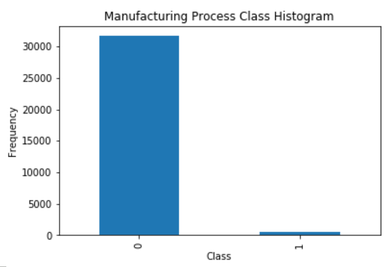
Introduction:

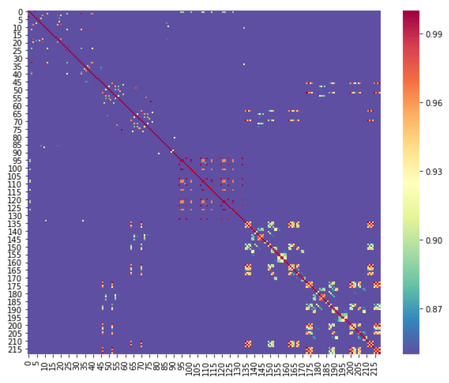
- Component selection through a series of tests is critical in the manufacturing process of technology components
- This selection process requires the components to pass a series of "specs“
- Each components' specs are encoded in a process parameter matrix
- Relying solely on the process parameter matrix of components is not enough to approximate the time to failure (a critical metric for this technology)
- We propose the use of Machine Learning to classify failure modes based on prior process parameters as features



Data Analysis:

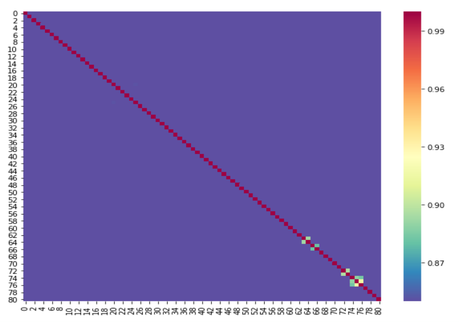
- Data sample size: 31,119
- 219 continuous Features
- Binary classification: two classes(“0” or “1”)
- Highly unbalanced: “1” to “0” ratio of 1.6%
- Highly correlated features

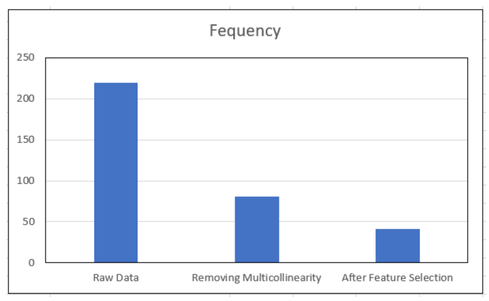




Data Preparation:

- Delete missing values
- Remove multicollinearity with Variance Inflation Factor (threshold = 10)
- Feature selection with ExtraTreesClassifier





Modeling:

- **Stacked Ensemble Method**

1. Partition the training data into 3 test folds:

ID	FoldID	X1	X1	X3	Y
1	2	XX	XX	XX	0
2	1	XX	XX	XX	0
3	3	XX	XX	XX	1
4	1	XX	XX	XX	0
5	3	XX	XX	XX	1

2. Create datasets train_meta and test_meta:

Train_meta

ID	FoldID	X1	X1	X3	M1	M2	M3	Y
1	2	XX	XX	XX	NA	NA	NA	0
2	1	XX	XX	XX	NA	NA	NA	0
3	3	XX	XX	XX	NA	NA	NA	1
4	1	XX	XX	XX	NA	NA	NA	0
5	3	XX	XX	XX	NA	NA	NA	1

Test_meta

ID	X1	X1	X3	M1	M2	M3	Y
6	XX	XX	XX	NA	NA	NA	0
12	XX	XX	XX	NA	NA	NA	0
22	XX	XX	XX	NA	NA	NA	1
75	XX	XX	XX	NA	NA	NA	0
88	XX	XX	XX	NA	NA	NA	0

3. For each test fold, combine the other two test folds to be used as train folds.

4. For each base model(M1: logistic regression, M2: AdaBoostClassifier, M3:Neural Network), fit the base model to the train folds and make predictions on the test fold. Store predictions in train_meta to be used as features for stacking model.

Train_meta

ID	FoldID	X1	X1	X3	M1	M2	M3	Y
1	2	XX	XX	XX	NA	NA	NA	0
2	1	XX	XX	XX	xx	xx	xx	0
3	3	XX	XX	XX	NA	NA	NA	1
4	1	XX	XX	XX	xx	xx	xx	0
5	3	XX	XX	XX	NA	NA	NA	1

5. Fit each base model to the full training dataset and make predictions on the test dataset. Store these predictions inside test_meta.

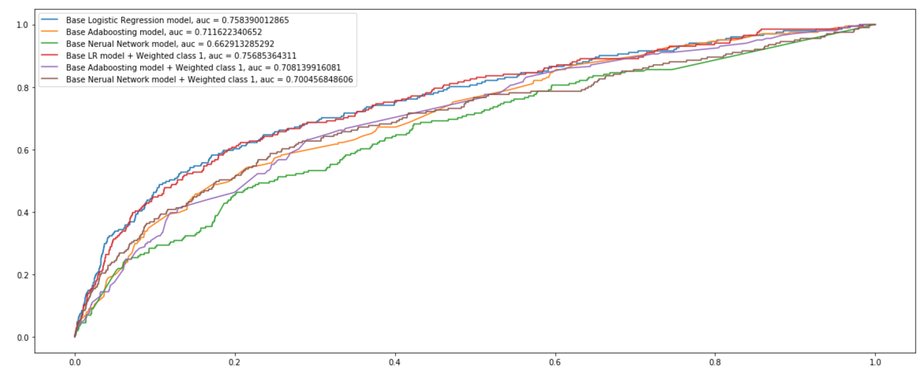
Test_meta

ID	X1	X1	X3	M1	M2	M3	Y
6	XX	XX	XX	xx	xx	xx	0
12	XX	XX	XX	xx	xx	xx	0
22	XX	XX	XX	xx	xx	xx	1
75	XX	XX	XX	xx	xx	xx	0
88	XX	XX	XX	xx	xx	xx	0

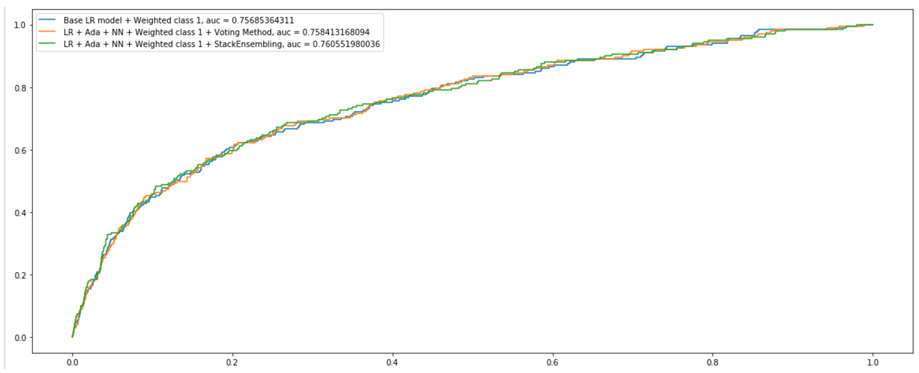
3. Fit a new model S on the train_meta and use the Stacked model to make final predictions on test_meta

Results & Evaluation:

- Logistic Regression shows better performance than Neural Network and AdaBoostClassifier
- Incrementing class “1” sample weight does not help on performance



- Stacked Ensemble & Bagging Method also do not help performance



- Supervised Learning vs Unsupervised Learning

			Predicted Condition 1	Predicted Condition 0	Precision	Recall
Supervised Learning	Logistic Regression	Condition 1	109	92	5.20%	54.20%
		Condition 0	2002	10245		
	Stacked Ensemble	Condition 1	58	143	6.20%	28.86%
		Condition 0	878	11369		
Unsupervised Learning	Novelty Detection(OneClassSVM)	Condition 1	N/A	N/A	9.50%	64.00%
		Condition 0	N/A	N/A		
	Outlier Detection(Isolation Forest)	Condition 1	26	175	4.10%	12.90%
		Condition 0	602	11645		
	Outlier Detection(Local Outlier Factor)	Condition 1	35	166	2.80%	17.40%
		Condition 0	1210	11037		

Conclusion:

- Quite clearly, the combination of high dimensions, small sample size, and high imbalance creates a difficult case for both supervised learning methods and unsupervised learning methods.
- Based on the Precision/Recall trade off, Logistic Regression shows the best Precision-Recall performance among Supervised Learning methods.
- Based on the precision/recall trade off, Novelty Detection/OneClassSVM shows the best Precision-Recall performance among Unsupervised Learning methods.
- To implement into the practical manufacturing process, Novelty Detection is the best choice. Although many good components maybe overkilled, 60% of the failed components will be caught before shipping to customers.