



电子科技大学
格拉斯哥学院
Glasgow College, UESTC

Final Year Project Report
Bachelor of Engineering

**Machine Learning-based Improvement of
Manufacturing Yield Rate**

Student: Jingyi Ran


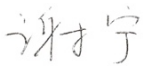

GUID: 2357641r

1st Supervisor: Ning Xie

2020–21

Coursework Declaration and Feedback Form

The Student should complete and sign this part

| | |
|---|--|
| Student Name: Jingyi Ran | Student GUID: 2357641r |
| Course Code : UESTC4006P | Course Name : INDIVIDUAL PROJECT 4 |
| Name of 1 st Supervisor: Ning Xie | Name of 2 nd Supervisor: Joao Ponciano |
| Title of Project: Machine Learning-based Improvement of Manufacturing Yield Rate | |
| Declaration of Originality and Submission Information | |
| <p><i>I affirm that this submission is all my own work in accordance with the University of Glasgow Regulations and the School of Engineering requirements</i></p> <p>Signed (Student) : </p> <p><i>I affirm this submission is completed by the student independently and the quality of the submission meets the requirements for graduation. I consent to the student taking part in the oral presentation.</i></p> <p>Signed (1st Supervisor) : </p> |  UESTC4006P |
| Date of Submission: Apr. 22th | |
| Feedback from Lecturer to Student – to be completed by Lecturer or Demonstrator | |

| | |
|--|--|
| <p>Grade Awarded:</p> <p>Feedback (as appropriate to the coursework which was assessed):</p> | |
| <p>Lecturer/Demonstrator:</p> | <p>Date returned to the Teaching Office:</p> |

Abstract

This project aims at researching on machine learning theory applied for data mining in the industry production line, especially for the complex data analysis of industrial field production process in discrete manufacturing industry. In ipad production line, there is a problem called “repeated testing”, which refers that one product needs to be tested whether it is pass or fail for several times because the testing status is different in every testing, and it needs multiple manual testing to determine the final status of the product. In this case, time and labor force would be wasted. In my project, I applied cutting-edge machine learning algorithm to do dichotomy for these products (whether they are pass or fail) based on their mass data in the production line.

To deal with mass industrial data, *featurization* and *classifier model training* are two key parts in this project. As for featurization part, **SMOTE** (Synthetic Minority Oversampling Technique) is used for data balancing; **Polynomial Features** and **Interaction Features** are used for feature dimension raising; **Correlation matrix** and **Chi square statistics** are used for feature selection;

As for classifier model training part, **LR** (Logistic Regression), **KNN** (K-NearestNeighbor), **SVC** (Support Vector Classification) and **DT** (Decision Tree) are applied to the datasets. In order to reduce the influence of overfitting, we apply K-fold cross-validation to get the final results.

KNN and SVC are best out of these classifiers according to the experiment results. Finally, our models achieve above 90% precision and recall for classification, which make great contribution to reduce “repeated testing” in industrial production line. In addition, since unbalancing categories are one of the most important features of the datasets in this project, an ablation experiment was carried out to verify the significance that SMOTE algorithm’s power in balancing positive and negative categories.

Acknowledgements

Firstly, I am very grateful to my supervisor Prof. Ning Xie during my work in this project and my writing in this report. Prof. Xie provided me help in learning basic theory as well as experiment setup. In his lab, I started to explore the world of data science and machine learning, which laid a good foundation for my future study in the field of data science.

In addition, I appreciate the support that Glasgow College, UESTC has given to me these years. With the outstanding education provided by Glasgow College, UESTC, I became a better student in academic and I gain a promising future in my further study.

Finally, during my work of the project, I met some outstanding researchers and senior students in the lab. I am very grateful to Yunfei Du, Xin Yan, who are research assistants in the lab. They offered great help in my research. Their patience, hard-working attitude and outstanding research outcome impressed me and inspire me to work harder in my research.

Contents

| | |
|---|----|
| Abstract..... | 4 |
| Acknowledgements..... | 5 |
| 1 Introduction..... | 7 |
| 1.1 Research Significance..... | 7 |
| 1.2 Related Work..... | 7 |
| 1.3 Motivations..... | 8 |
| 2 Implementation..... | 9 |
| 2.1 Proposed Model..... | 9 |
| 2.2 Database and Data Cleaning Outcomes..... | 9 |
| 2.3 Featurization..... | 10 |
| 2.3.1 Balancing dataset: Smote Algorithm..... | 10 |
| 2.3.2 Feature Dilation..... | 12 |
| 2.3.3 Normalization..... | 14 |
| 2.4 Classifier..... | 14 |
| 2.4.1 KNN classifier..... | 14 |
| 2.4.2 SVM classifier..... | 15 |
| 2.4.3 Logistic Regression..... | 16 |
| 2.4.4 Decision Tree..... | 17 |
| 3 Result and Analysis..... | 18 |
| 3.1 Model Evaluation Index..... | 18 |
| 3.2 Experiment result:..... | 20 |
| 3.2.1 OSD-Module: | 20 |
| 3.2.2 WIFI-Module..... | 21 |
| 3.3 Result Analysis:..... | 22 |
| 3.3.1 OSD-Module..... | 22 |
| 3.3.2 WIFI-Module..... | 23 |
| 3.4 Ablation Analysis..... | 23 |
| 4 Conclusions and Future Work..... | 25 |
| References..... | 27 |

1 Introduction

1.1 Research Significance

This project aims at applying machine learning-base data mining skills to analyze mass data from industrial production line and thus improve the manufacturing yield rate. Through the study, in discrete manufacturing field, artificial intelligence will boost the intelligence process of industrial production, which will significantly improve the accuracy and efficiency of intelligent manufacturing.

Manufacturing used to rely on electricity, however, with the trend of artificial intelligence, future manufacturing will rely on big data. The development of Internet of Things, Big Data and the association between the Internet and manufacturing inspire us how to optimize the process of manufacturing, like improving Data control of manufacturing process, predicting equipment failur and upgrading production line. Apply machine learning to improve the manufacturing yield rate has significant values.

1.2 Related Work

The development of automatic detection system faces many problems because of the current lack of data mining-based fault diagnosis and prediction method for automatic detection system. Fault diagnosis and prediction in other fields have the related research. Based on data to do specific fault diagnosis, researchers in Aberdeen University developed The TIGER system for turbine fault diagnosis based on data mining technology[1]; Li Hailin et al. designed an algorithm for detecting engine faults by using the morphological characteristics of engine parameters and the time series method[2];

Recent years, more fault diagnosis methods based on machine learning algorithm came to the industry field. According to the characteristics of discrete data, Zhou Hao designed the way and method to extract the system state features, and used decision tree to learn fault rules from the analyzed data[3]; Cheng Huali et al. applied machine learning method to the detection of motor noise according to the characteristics of motor sound signals, thus reducing the difficulty of manual inspection [4]. Dong Yong et al. used BP neural network, SVM, decision tree and simple Bayes to carry out experiments on disk fault prediction, compared the prediction time window and accuracy of the four methods under this problem, and pointed out that the neural network method has a strong dependence on parameters. In the review of the correlation vector machine in fault prediction, Ma Dengwu et al. pointed out that the intelligence of the model

constructed by this kind of machine learning method is the development direction of the future fault diagnosis field, and different kinds of methods can complement each other [5].

The problem of fault prediction for industrial systems is also based on analyzing the correlation between operational data and specific fault types. Therefore, the same method can be migrated to the problem of fault prediction for industrial systems.

1.3 Motivations

The ultimate purpose of this research is to train the machine learning model of supervised or unsupervised classification through machine learning method, develop the industrial defective classification algorithms, and the final result is the trained model for this problem. In order to achieve this goal, the key issues to be solved are as follows:

The same workpiece, the station repeated test problem:

By analyzing raw data from the production line, there exists repeated tests. The problem is that the test equipment cannot accurately judge the status of the assembled workpiece (such as MIC, WIFI module, screen display, etc.). For example, the test status of same workpiece after one Pass will become Fail, and the test status of same workpiece after one Fail may be Pass. Then the status of this workpiece will be repeatedly tested again. This will inevitably cause some waste of time costs and labor force cost.

Assume dataset is $D=\{d_1, d_2, d_3...\}$; feature is $X = \{x_1, x_2, x_3...\}$, $\{d_1 = x_1, d_2 = x_2, d_3 = x_3...\}$, detection status is $T = \{\text{Pass}, \text{Fail}\}$, detection device is `raw_machine()`. In this case, the above problem could be described as: If d_1 is a repeated test data, it tests three times, the first two status are Fail and the final status is Pass, that is, the final test state of d_1 is specified as Pass, then our ultimate goal is to test d_1 only once and the accuracy of test state is above 70% Pass.

Test_1 $d_1: \text{Fail} = \text{raw_machine}(x_1)$

Test_2 $d_1: \text{Fail} = \text{raw_machine}(x_2)$

Test_3 $d_1: \text{Pass} = \text{raw_machine}(x_3)$

As shown in the above formula, the state of d_1 is repeatedly detected three times, which wastes the time and labor force.

2 Implementation

2.1 Proposed Model

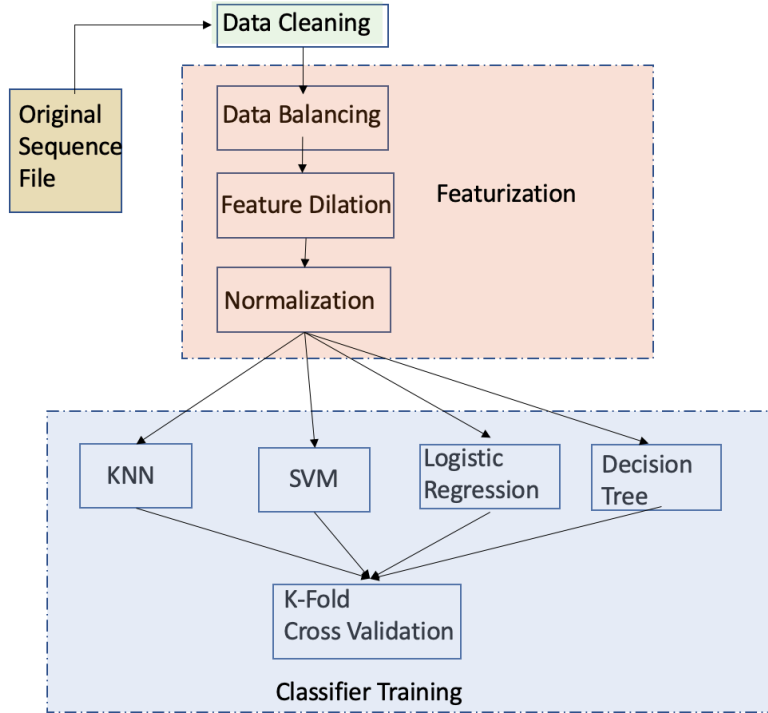


Fig.1: Proposed model for the project

2.2 Database and Data Cleaning Outcomes

Data cleaning part was finished by applying *Numpy and Pandas*, the programming language is *Python3*.

There are two databases used in this project: Wifi Module of iPad production line and OSD Module of iPad production line. The dataset and parameters are shown in Table 1:

Table 1: Dataset detail (Original Dataset)

| <i>Dataset</i> | <i>Parameters Specifications</i> | <i>Proportion of Pass/Fail Status</i> |
|--------------------|----------------------------------|---------------------------------------|
| <i>Wifi-Module</i> | Parametric: 12-dimensions | Pass : Fail=26118:1727 |
| <i>OSD-Module</i> | Parametric: 8-dimensions | Pass : Fail=393401:13711 |

Parameters in the datasets are sampling values in same intervals of reflection frequencies when the Module is testing.

In this project, the key problem is: at the same workpiece, the station repeated test problem. Assume dataset is $D=\{d_1, d_2, d_3 \dots\}$; testing status is $T = \{\text{Pass}, \text{Fail}\}$, in this case, the problem could be described as: If d_1 is a repeated test data, it tests three times, the first two status are

Fail and the final status is Pass, that is, the final test state of d_1 is specified as Pass, then our ultimate goal is to test d_1 only once and the accuracy of test state is above 70% Pass.

To solve this problem, I chose to retain only the last testing data of every product. At data cleaning stage, data record that are not an indication of last testing will be deleted.

After data cleaning, the dataset and parameters are shown in Table 2:

Table 2: Dataset detail after data cleaning

| Dataset | Parameters Specifications | Proportion of Pass/Fail Status |
|-------------|---------------------------|--------------------------------|
| Wifi-Module | Parametric: 12-dimensions | Pass : Fail=124807:19 |
| OSD-Module | Parametric: 8-dimensions | Pass : Fail=382872:1939 |

2.3 Featurization

The key significance of featurization is to transform original data to features needed by machine learning classification algorithms.

2.3.1 Balancing dataset: Smote Algorithm

In this section we mainly covered 3 parts: 1) why balancing algorithm is mandatory for this project; 2) an introduction to SMOTE; 3) the visualization results of SMOTE.

One disadvantage of datasets in this project is extremely unbalancing. For example, the Pass/Fail distribution of Wifi-Module dataset is shown below in Fig.1:

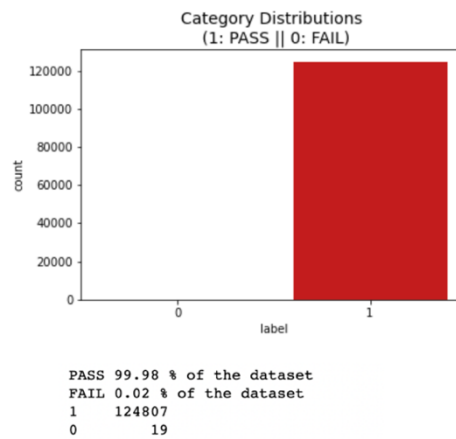


Fig.1: Class Distribution of Wifi-Module

According to the result above, we can find that the proportion of two classes are extremely unbalancing. If we directly input the dataset into classifier, the prediction results are also often unreliable, which means the classification results will be towards categories which have more

amounts, in this project, Pass is the more categories. For example, in the Wifi-Module, to reach the highest accuracy, the classifier could make all samples as “Pass”, in this case, the accuracy will reach 99.98%, however, this classifier is randomly making predictions and it has not use.

To balance the dataset, we chose to apply SMOTE (Synthetic Minority Oversampling Technique) to the dataset, and try to make the dataset balanced. After SMOTE, a few category samples are analyzed and simulated, and new samples of manual simulation are added to the data set, in this case, the imbalance of category of Pass and Fail is not a problem any more.

The SMOTE principle and process are shown below:

For each sample X in the minor category, Euclidean distance is used to calculate the distance from it to all samples in the dataset of the minority class, and in this case, the k nearest neighbor will be get.

According to the sample imbalance ratio, a sampling ratio is set to obtain the sampling ratio N . Number of samples are randomly selected from its k nearest neighbors for each minority sample x , and the selected nearest neighbor is set to be x_n .

A new sample is constructed respectively with the original sample for each randomly selected neighbor x_n , the related formula is shown below:

$$x_{smote_new} = x + rand(0,1)(\tilde{x} - x)$$

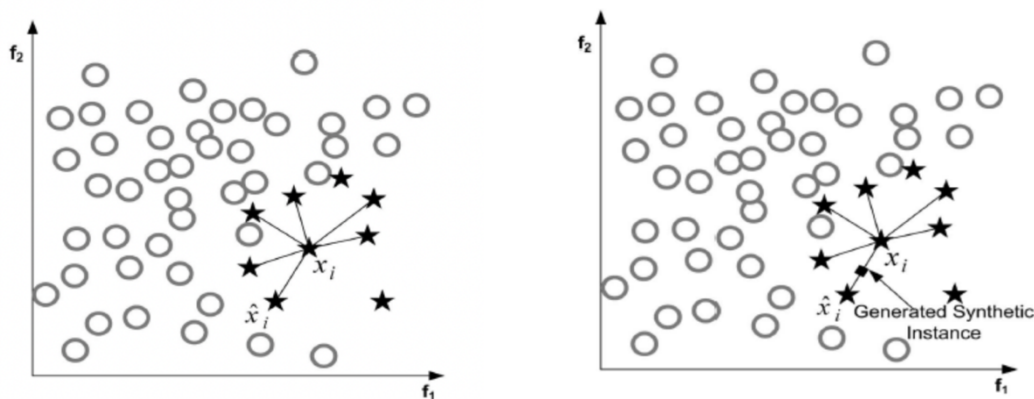


Fig.2: Working principle of Smote algorithm

After applying Smote algorithm, the category distribution became balanced. The Pass/Fail distribution of Wifi-Module dataset is shown below:

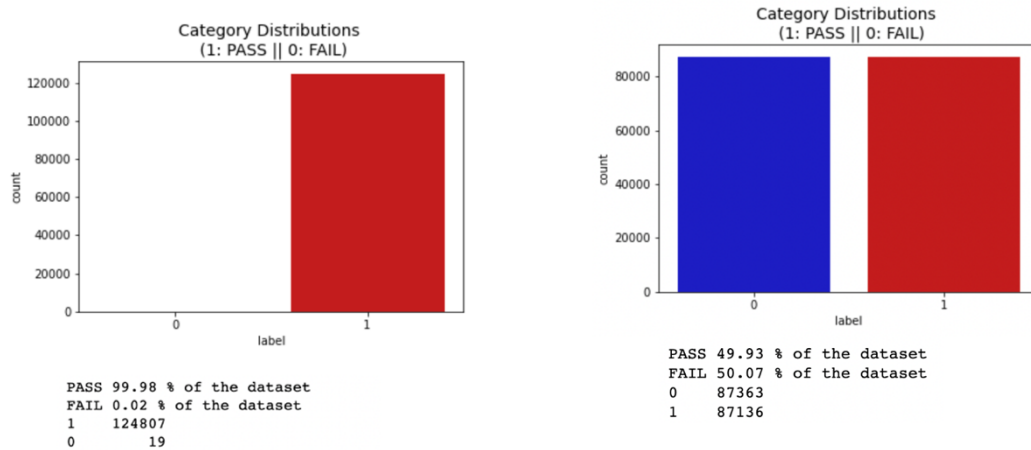


Fig.3: Comparison of category distribution before and after Smote analysis

Fig.3 shows the comparison of feature distribution before and after Smote. By drawing the scatter diagram of features in the dataset, we could find that before smote, almost all areas are Pass data, in this case, the classifier is easily to ignore Fail data since they are so few. After applying Smote, we could see the distribution of two categories became balanced.

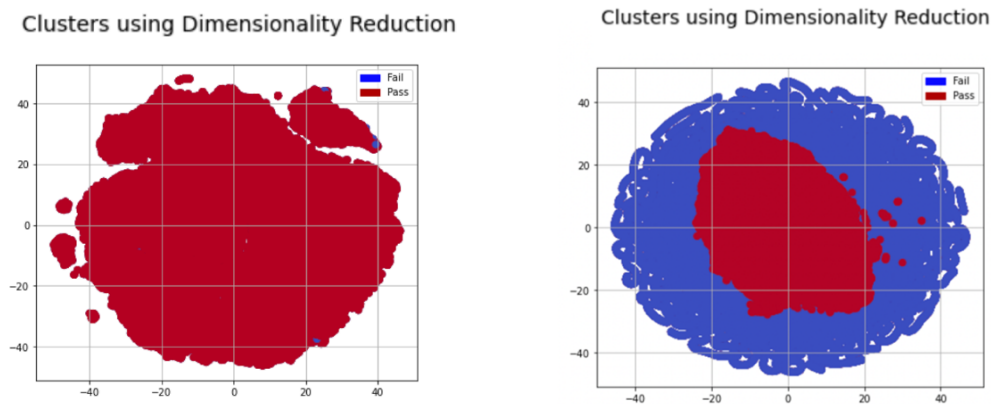


Fig.4: Comparison of feature distribution before and after Smote analysis

An ablation experiment was carried out and the result is shown in Part 3.4 Ablation Analysis.

2.3.2 Feature Dilation

In the original dataset, the features are 12-dimensions for WIFI-module and 8-dimensions for OSD-Module, which is not enough for model training. In my experiment, the model shows underfitting, which means the model complexity is not enough. In this part, I applied two dimension rising methods to the dataset: Polynomial Feature and Interaction Feature.

2.3.2.1 Polynomial Feature

When the relationship between features and labels are not linear, polynomial feature is needed. For example, age may be closely related to health, and health condition will be worse when age increases. However, their relationship is not linear. In this case, the feature X needs to be coded to generate a higher order form of the feature (x^2, x^3) to represent the nonlinear effect on the label.

The parameter “degree” will decide the highest order of polynomial. For example, degree=2 will create features with highest order 2:

$$x_1, x_2, x_1^2, x_2^2$$

degree=3 will create features with highest order 3:

$$x_1, x_2, x_1^2, x_2^2, x_1^3, x_2^3$$

In this dataset, original features are sampling values in same intervals of reflection frequencies when the Module is testing. According to these sampling values, the labels (Pass / Fail) are decided. It is clear that the relationship between features and labels are not linear.

After applying polynomial Feature, the dimensions of the features in two dataset are shown below:

Table 3: Feature dimensions after dilation

| <i>Dataset</i> | <i>Parameters Specifications</i> |
|--------------------|----------------------------------|
| <i>Wifi-Module</i> | Parametric: 219-dimensions |
| <i>OSD-Module</i> | Parametric: 329-dimensions |

Although adding polynomial features will be helpful for improving the model complexity, it is not mean the more polynomial features to be added, the better the model will be. Since the polynomial features are associated, if we add to many polynomial features, we will have redundant features, which is not efficient for model training.

In the experiments, we adjusted the parameter “degree” to find the best parameter (according to the accuracy of SVM classifier)

Table 4: Accuracy versus degree parameter in Polynomial

| <i>Dataset</i> | <i>degree=1</i> | <i>degree=2</i> | <i>degree=3</i> | <i>degree=4</i> | <i>Degree=5</i> |
|----------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| <i>OSD-Module Accuracy</i> | 79.6% | 82.3% | 85.71% | 85.69% | 85.72% |

Therefore, according to the experiment results, the features are 219-dimensions for Wifi module and 329-dimensions for OSD-Module.

2.3.2.2 Interactive Features

In addition, sometimes we have a situation where one feature depends on another in order to have an effect on the label. To predict whether a cup of coffee is sweet, for example, consider two characteristics: whether the coffee has been stirred and whether the coffee has been sweetened. Looking at one of the two alone does not confirm that the coffee is sweet, but together they do. Coffee is sweet only when it is sweetened and stirred, and the effect of these two characteristics on the label (coffee is sweet) is interdependent. By generating an interaction feature, we can increase the complexity of the model

In this dataset, original features are sampling values in same intervals of reflection frequencies when the Module is testing. Therefore, it is clear that the features are dependent.

2.3.3 Normalization

Normalization is a process of transforming original data into new data whose values are from 0 to 1. The formula of normalization is shown in formula 1:

$$Feature_{norm} = \frac{(Feature - Min(Feature))}{Max(Feature) - Min(Feature)}$$

Normalization is to eliminate unnecessary error caused by data's different type. The key of normalization is one kind of linear transformation, which means after normalization, the sort of original data will not be changed.

Two important classifiers in this project are KNN and SVC, and both of them need normalization. For KNN, the classifier will calculate distances between samples, like Euclidean Distance. If the range of a feature value is very large, then the distance calculation depends mainly on this feature, which is contrary to the actual situation. For SVM, when using Gaussian kernel, it assumes all dimensions depends on one variance, which require the distribution of features input the classifier to be the same.

2.4 Classifier

2.4.1 KNN classifier

KNN algorithm is an unsupervised machine learning method. KNN algorithm is on-parametrim, which means no pre-assumptions during data training, in this case, the training process becomes more independent of the data distribution. In addition, KNN is also instance-based (model's prediction result only relies on the other training examples [6]).

Three main approaches of calculating the distance between the need-to-predict observations to all labeled observations are Euclidean distance, Manhattan distance and Minkowski distance. K is a very important parameters, if we chose K small, overfitting phenomenon will appear; If K becomes larger, the smoother the boundary between various class will be.

In this project, we choose K to be 5. More detailly, Euclidean distance will judge the category of each point considering 5 neighboring points. The formula is shown below:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

In addition, to reduce the influence caused by overfitting and to make the experiment result better, cross-validation is added to the K - fold validation method. Generally, the K value for cross-validation is chosen to be 1 to 5. We set the K value to be 3 in experiments. To be specific, the dataset was split into 3 folds, and each one will be used as testing set, and the others will be used as training set. Finally, the accuracy, precision, recall as well as f1-score will show average value of each training.

2.4.2 SVM classifier

linear separability

Support Vector Machine (SVM) is a linear classifier which is one of supervised learning methods. The decision boundary of SVM is a maximum-margin hyperplane for training samples. Hinge loss function was used to evaluate risk. In addition, SVM could carry out nonlinear classification by using kernel method

The learning objective is split by positive and negative categories, and the distance between the point to the plane of any sample is larger than or equal to 1.

Decision boundary: $w^T X + b = 0$

Point to plane distance: $y_i(w^T X_i + b) \geq 1$

The decision boundary forms two parallel hyperplanes:

$$w^T X_i + b \geq +1 \quad y_i = +1$$

$$w^T X_i + b \leq -1 \quad y_i = -1$$

Above the upper interval boundary, samples belong to the positive category; Below the lower interval boundary, samples belong to the negative category. Margin is the distance between two interval boundaries $d = \frac{2}{||w||}$ [7].

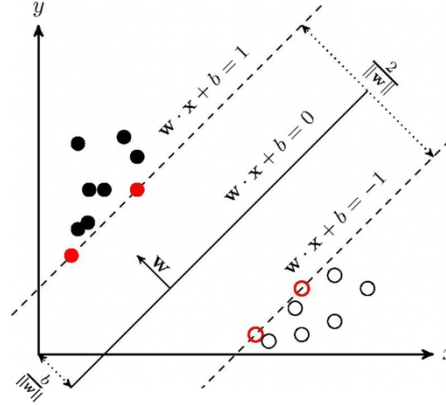


Fig.5: Working principle of SVM

Common Kernel Function

Some examples of kernel functions are given below:

Table 5: kernel functions in SVM

| Name | Analytic Expression |
|-------------------|--|
| Polynomial kernel | $\kappa(\mathbf{X}_1, \mathbf{X}_2) = (\mathbf{X}_1^\top \mathbf{X}_2)^n$ |
| RBF kernel | $\kappa(\mathbf{X}_1, \mathbf{X}_2) = \exp\left(-\frac{\ \mathbf{X}_1 - \mathbf{X}_2\ ^2}{2\sigma^2}\right)$ |
| Laplacian | $\kappa(\mathbf{X}_1, \mathbf{X}_2) = \exp\left(-\frac{\ \mathbf{X}_1 - \mathbf{X}_2\ }{\sigma}\right)$ |
| Sigmoid kernel | $\kappa(\mathbf{X}_1, \mathbf{X}_2) = \tanh[a(\mathbf{X}_1^\top \mathbf{X}_2) - b], \quad a, b > 0$ |

In this project, we choose RBF kernel to do the classification, the result will be shown in result analysis part.

2.4.3 Logistic Regression

Logistic regression has much in common with multiple linear regression analysis. The dependent variables of logistic regression can be dichotomous or multi-classification, but dichotomous is more commonly used and easier to explain, and multi-classification can be processed by Softmax method. In practice, the most commonly used is the binary logistic regression.

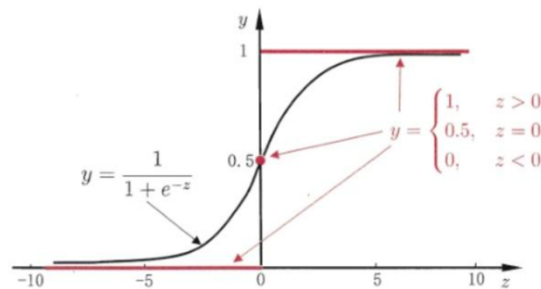


Fig.6: Sigmoid Function

3.2.4 Decision Tree

Decision tree is a tree-shaped structure where each internal node represents a judgment on an attribute, and each branch shows the output of a judgment result.

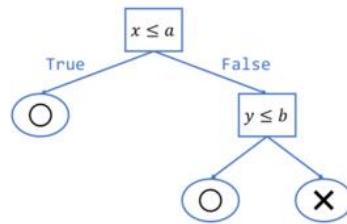


Fig.7: Classification principle in decision tree

The parent node of decision tree is determined by Entropy principle. For a dataset, we hope the entropy to be lower, in this case, we can get the better classification result.

$$\text{Entropy} = - \sum [p(x_i) * \log_2(P(x_i))]$$

P of x_i is occurrence probability of x_i . When there's 50% of A and 50% of B,

$$\text{Entropy} = - (0.5 * \log_2^{0.5} + 0.5 * \log_2^{0.5}) = 1$$

When there's only category A, or when there's only category B,

$$\text{Entropy} = - (1 * \log_2^1 + 0) = 0$$

So, when Entropy is at most 1, it is the least effective state for classification, and when it is at least 0, it is the state for complete classification. Because entropy equals zero is ideal, and in general, in real life, entropy is somewhere between zero and one.

3 Result and Analysis

The results are tested based on *Jupyter Notebook*. Programming language is *Python 3*. The results and analysis are given in this chapter.

3.1 Model Evaluation Index

Experiment results are shown by accuracy, precision, recall, f1-score, roc-auc and learning curve respectively.

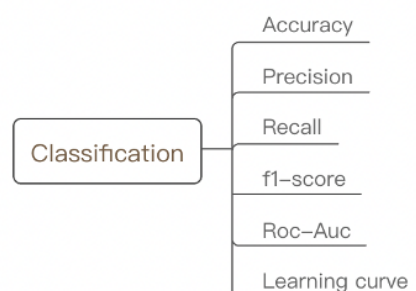


Fig. 8: Model evaluation matrix

Confusion Matrix:

True Positive(TP): Classify positive to positive

True Negative(TN): Classify negative to negative

False Positive(FP): Classify negative to positive (Type I error)

False Negative(FN): Classify positive to negative (Type II error)

Table 6: Relationship between true value and predicted value

| Real Category | Predictive Category | | | |
|------------------|---------------------|-----|----|-------|
| | | Yes | No | Total |
| | Yes | TP | FN | P |
| | No | FP | TN | N |
| | Total | P' | N' | P+N |

Accuracy: $ACC = (TP+TN)/(TP+TN+FP+FN)$

Error rate: $error\ rate = (FP+FN)/(TP+TN+FP+FN)$

Sensitive: $sensitive = TP/P$

Precision: $\text{precision} = \text{TP} / (\text{TP} + \text{FP})$

Recall: $\text{recall} = \text{TP} / (\text{TP} + \text{FN}) = \text{TP} / \text{P} = \text{sensitive}$

F-Measure: $\text{F1} = 2\text{PR} / (\text{P} + \text{R})$

In machine learning, one application is cancer prediction. According to patients' data, model needs to predict whether the patient gets cancer or not. In this case, there is a preference. If the patient does not get cancer, but the model predicts him get cancer, the patient will waste time to affirm; In addition, if the patient gets cancer, but the model predicts him not get cancer, the situation is serious, which means a cancer patient does not know he gets cancer and he may miss the chance to do medical examination.

This example shows that the model prediction should have preference. Firstly, the model must make correct prediction for those who get cancer, then, the model should try best to make correct prediction for those who do not get cancer to reduce time waste.

In this project, the model also has preference. We paid more attention to the Fail data, which means we want the model to make correct predictions for Fail data as much as possible, in this case, we can avoid the Fail product predicted to be Pass, and avoid customers get the Fail product. If the model makes wrong predictions for Pass product (Pass product predicted to be Fail), it will cost time waste to test again, but it will not cause very serious problem.

To solve the problem, we paid special attention to one parameter in SVM model: `class_weight`. Since in the dataset, Fail category only counts for small part of the whole dataset, we set this parameter to make the model is modeled in the direction of capturing a few classes. In general, this parameter is set to None, however, in this project, we adjusted this parameter:

```
from sklearn import svm
clf = SVC(C=1.0, break_ties=False, cache_size=200, class_weight={0:0.3, 1:0.7}, coef0=0.0,
        decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
        max_iter=-1, probability=False, random_state=None, shrinking=True,
        tol=0.001, verbose=False)
clf.fit(X_resampled, y_resampled)
#kernel='rbf'
```

Fig.9: Parameter choice in SVM

The proportion of Pass data and Fail data input to the model is not 1:1, instead, we set this proportion to be 0.3:0.7, more Fail data will be input to make the model realize the preference for Fail data.

In addition, as for the evaluation index, we pay more attention on recall. The formula of recall is shown below:

$\text{recall} = \text{TP} / (\text{TP} + \text{FN}) = \text{TP} / \text{P} = \text{sensitive}$



Fig.10: Two type of errors

Therefore, our task is transferred: the recall must be high (we cannot stand that Fail products are predicted to be Pass); after achieving that, try making the precision high (save repeating test time for those Pass products predicted to be fail).

3.2 Experiment result:

In the experiment, I applied four classifiers to two datasets: OSD-Module and WIFI-Module. The experiment results are shown in this part:

3.2.1 OSD-Module:

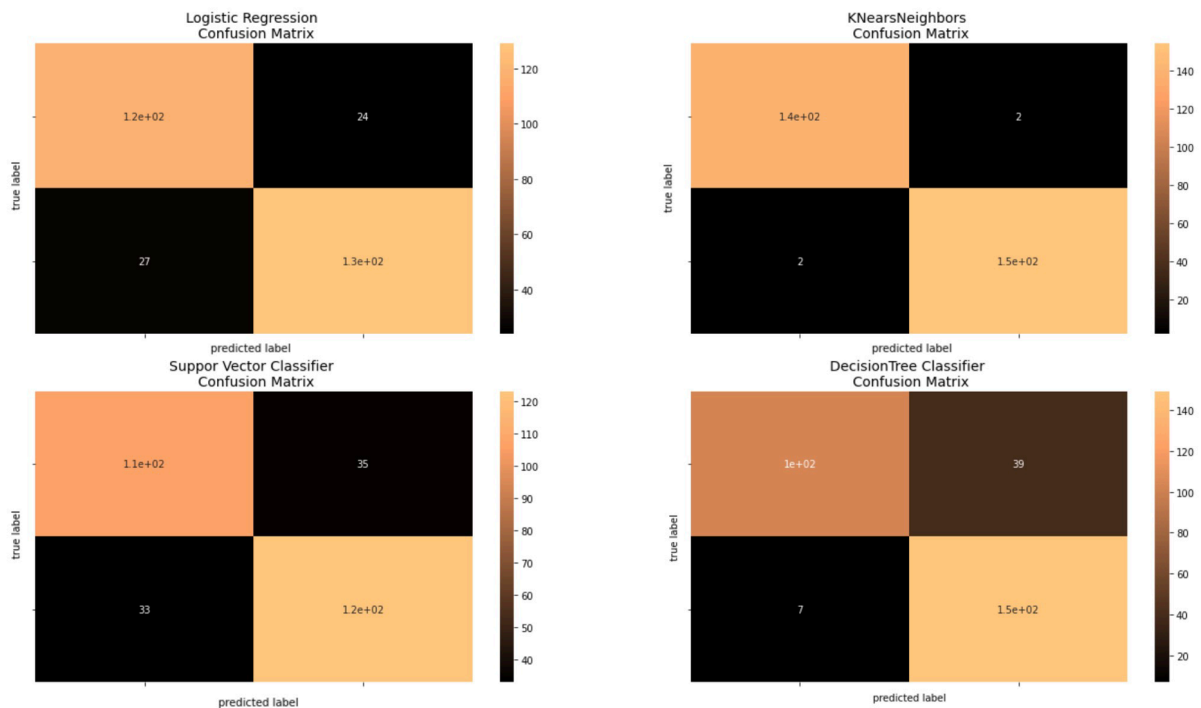


Fig.10: OSD-Module Confusion matrix for four classifiers

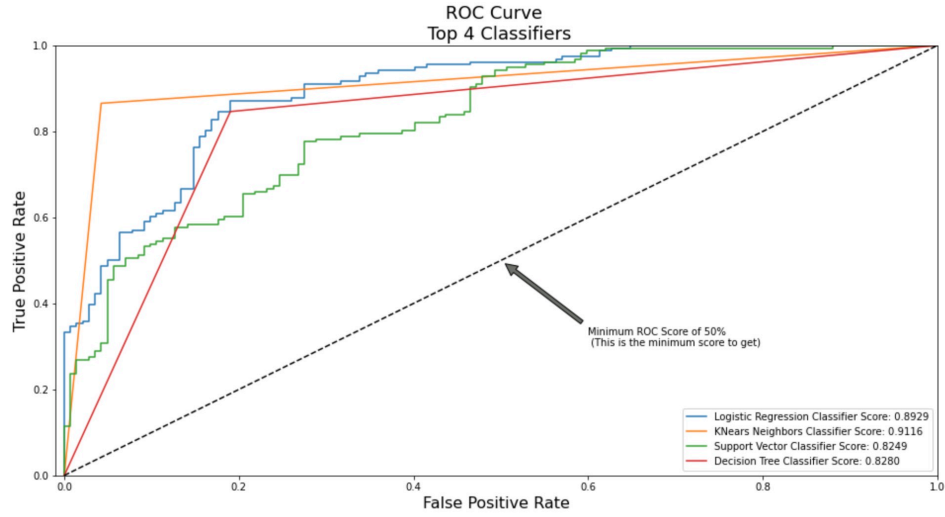


Fig.11: OSD-Module ROC curve for four classifiers

Table 7: OSD-Module Experiment result

| | Parameters | Recall | Precision | Accuracy | F1-score | Auc |
|----------------------------|------------------------------------|------------|------------|------------|------------|------------|
| Logistic Regression | C=10000; max_iter=10000 | 83% | 84% | 83% | 83% | 89% |
| KNN | n_neighbors=2 | 92% | 89% | 95% | 91% | 91% |
| SVM | C=1 | 81% | 84% | 80% | 81% | 82% |
| Decision Tree | max_depth=3; min_samples_leaf=6 | 91% | 87% | 90% | 91% | 82% |

(Parameters are obtained by *Verification curve* and *Grid Search*)

3.2.2 WIFI-Module

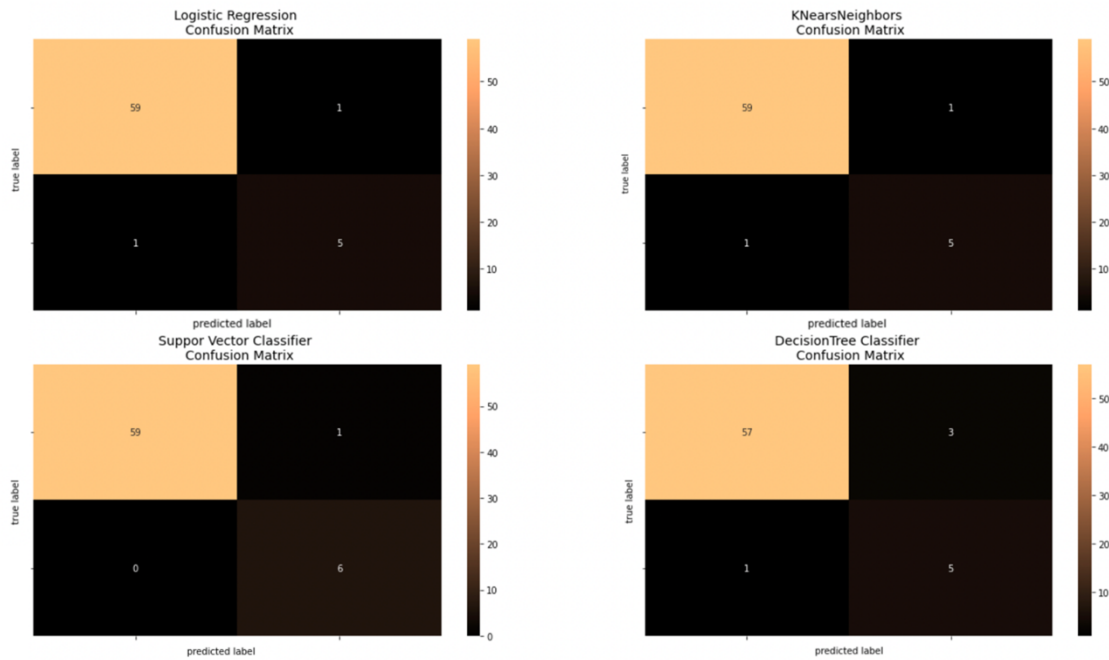


Fig.12: WIFI-Module Confusion matrix for four classifiers

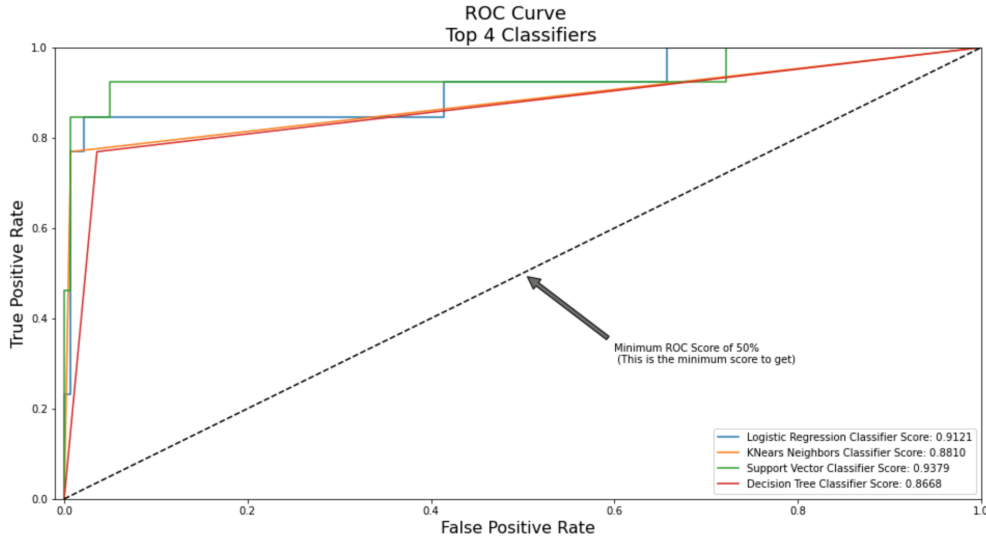


Fig.13: WIFI-Module ROC curve for four classifiers

Table 8:Wifi-Moule Experiment result

| | Parameters | Recall | Precision | Accuracy | F1-score | Auc |
|----------------------------|------------------------------------|------------|------------|------------|------------|------------|
| Logistic Regression | C=100; max_iter=10000 | 83% | 83% | 97% | 83% | 91% |
| KNN | n_neighbors=3 | 83% | 83% | 97% | 83% | 88% |
| SVM | C=1 | 94% | 86% | 98% | 92% | 93% |
| Decision Tree | max_depth=2; min_samples_leaf=5 | 83% | 62% | 94% | 71% | 86% |

(Parameters are obtained by *Verification curve* and *Grid Search*)

3.3 Result Analysis:

3.3.1 OSD-Module

In OSD-Module dataset, the results reflect that KNN performs best in large dataset (in this dataset, it is about 380,000). The reason behind that may due to well avoiding the decline of accuracy and insufficient generalization caused by overfitting.

Besides, KNN receives original features (without transformation) during the training process, and remains the dimension steady; however, some kernel classifiers like SVM maps the original features to a higher dimension. This proves that the original features are most suitable for this specific dataset. Therefore, for other datasets, KNN may not perform as well

as the other classifiers(Example is WIFI-Module dataset, which will be illustrated in detail in Part 3.3.2).

Results also show that when SVM classifier is applied, the recall declines by 15%, compared with KNN, 12% with Logistic Regression and 5% with Decision Tree. Therefore, we can draw the conclusion that KNN has advantages in the field of processing large-scale industrial data outliers.

3.3.2 WIFI-Module

In WIFI-Module dataset, the results reflect that SVM methods have the best performance. As SVM maps the dimension of the original features into a higher one to increase the feature complexity. As SVM is a nonlinear method actually, which shows that if the sample size is small, it could grasp the nonlinear relationship of data and features. However, if the sample size is not small, linear classification methods' disadvantage is smaller. For example, non-linear relationships can be simulated by manually splitting/discretizing features. At this point, the advantages of SVM are not obvious.

Therefore, for datasets simple-feature and small size dataset like WIFI-Module, SVM is more reliable compared with simple classifier like KNN and Logistic Regression.

3.4 Ablation Analysis

In this project, SMOTE algorithm is applied to balance dataset, the detail of SMOTE algorithm has been illustrated in Part 2.2. In this part, an ablation analysis will be introduced.

For classifier, the advantages of balancing dataset could be shown in ROC curve and AUC value. AUC is a performance index to measure the performance of a classifier. ROC is receiver operating characteristic curve. It is plotted on the vertical axis of true positive rate (sensitivity) and abscissa of false positive rate (specificity) according to a series of different dichotomies (cut-off values or determination thresholds). AUC is the area under ROC curve. The details about the performance index have been illustrated in Part 3.1.

After applying Smote analysis, AUC value increased significantly, which means Smote algorithm largely improve the performance of the classifiers.

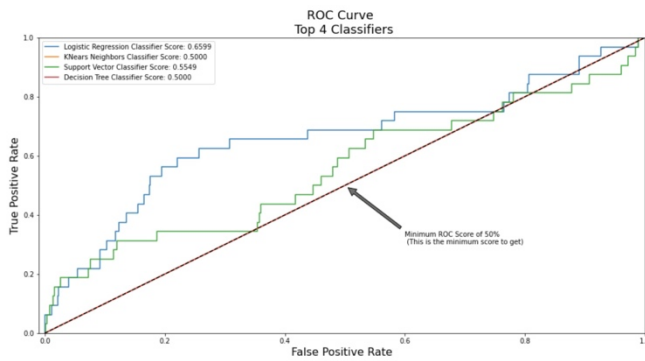


Fig.14: OSD-Module ROC curve before Smote

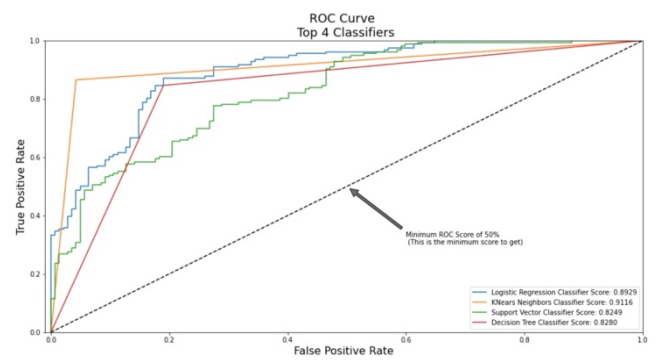


Fig.15: OSD-Module ROC curve after Smote

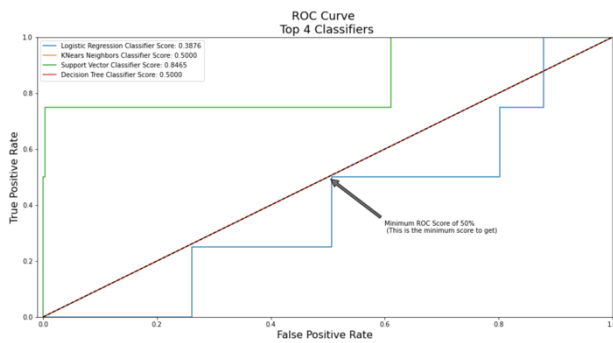


Fig.16: WIFI-Module ROC curve before Smote

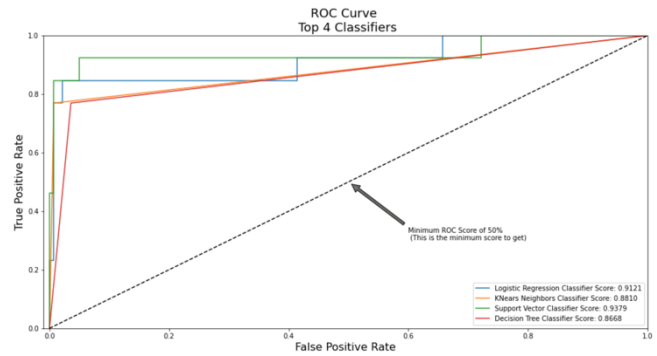


Fig.17: WIFI-Module ROC curve after Smote

4 Conclusions and Future Work

The goal of this project is to train a data-driven supervised/unsupervised model through frequently used machine learning methods used for defectiveness detection in industry scenarios. Finally, we used a set of persuasive and robust evaluation matrices to measure, estimate and judge the risk of defective products are realized during the manufacturing process, and thus effectively improve the productions' yield rate.

The two key parts of this project is **feature engineering** and **classifier training**.

(1). For feature engineering, this project proves the significance of SMOTE in balancing dataset, and largely improve the performance of the classifier when training extremely unbalanced dataset. In addition, this project verifies the importance of feature dilatation for improving feature complexity, by applying Polynomial Feature dilatation and adding interactive features, the feature complexity could be improved significantly, which will reduce overfitting.

(2). For classifier training, this project verifies the advantages of KNN model when dealing with large-scale industrial data outliers; however, trade-off should be considered in real industrial scenarios. To illustrate, KNN is a simple and efficient when dealing with large dataset to avoid the decline of accuracy and insufficient generalization caused by overfitting. In addition, we also verified SVM can get much better results than other algorithms on small size training sets. SVM uses complex kernels to solve the nonlinear classification by projecting features into a higher dimensional space and achieve great classification performance;

Other classifiers used in this project also have advantages: Decision Tree can generate understandable rules and work with contiguous and category fields. Logistic Regression is simple to implement, and very efficient (low computation, low storage footprint) to be used in big data scenarios.

According to the experiment results, one classifier might perform well in a specific evaluation matrix but another classifier performs well in another. The selection of classifier depends on the needs of project, in other words, the trade-off between accuracy, recall and precision, which will led to different choice of classifier.

Quantitatively, our model in this project could achieve 90% precision in classification for the manufacturing dataset, which could largely improve the efficiency in the manufacturing production line.

In the future, we plan to put forward innovative production process data (parts data, the production process, equipment operation sequence data and testing data) diversity

characteristics of integrated framework, further establish a unified semantic expression benchmark, provide a rich industries production process context and semantic consistency definition and feature description method. In the field of intelligent industrial process, realize the data-driven online production quality intelligent detection and anomaly analysis software system.

References

- [1] Q. L. Guo, H. F. Wu, Rough Set Data Mining and Its Application in Turbine Fault Diagnosis[J]. Modern electronic technology, 2006,29 (13):74-77.
- [2] H. L. Li, C. H. Guo, L. B. Yang, Fault diagnosis method based on time series data mining[J]. Data acquisition and processing, 2016, 31(4):782-790.
- [3] H. Zhou. Research on key technologies of fault prediction of E-class system based on machine learning [D]. National University of Defense Technology, 2011.
- [4] H. L. Cheng, K. Q. Fan, Machine learning method for abnormal tone detection and its application in motor quality inspection [J]. Measurement and control technology, 2015, 34(4):55-58.
- [5] G. B. Huang, M. B. Li, L. Chen, C. K. Siew, Incremental extreme learning machine with fully complex hidden nodes, Neurocomputing 71 (1) (2008) 380 4–6.
- [6] Zengming W, Kaiyu W, Improvement of K-nearest neighbor algorithm based on entropy weight, 2009, CNKI
- [7] Vapnik, V.. Statistical learning theory. 1998 (Vol. 3). . New York, NY: Wiley, 1998
- [8] Qin, J. and He, Z.S., 2005, August. A SVM face recognition method based on Gabor-featured key points. In Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on (Vol. 8, pp. 5144-5149). IEEE.
- [9] Saiyu C, The application of several decision probability models in real life, Journal of theory and practical theory, 2006 (5)
- [10] Ng R T,Han J. CLARANS:A method for clustering objects for spatial data mining[J]. IEEE T ransactions on K nowledge and Data Engineering ,2002,14(5) :1003- 1016.
- [11] Karpis G,Han E H, Kumar V. Chameleon: Hierarchical clustering using dynamic modeling[J]. Computer, 1999.32(8) :68-75.
- [12] Zhang T , Ramakrishnan R, Livny M. BIRCH: An efficient data clustering method for very large databases[C] // Proc of 1996 ACM SIGMOD International Conference on Manage-ment of Data,1996:103-114.
- [13] Wang Wei, Yang Jiong, Muntz R. A statistical information grid approach to spatial data mining[C]// Proc of the 23rd Conference on VLDB, 1997 :186- 195.
- [14] Pankaj Malhotra,Lovekesh Vig,Gautam Shroff,Puneet Agarwal. Long Short Term Memory Networks for Anomaly Detection in Time Series, ESANN 2015
- [15] JIA Peng-tao,HE Hua-can,LIU LI,SUN Tao. Overview of time series data mining, Application Research of Computers,2007,24(11)
- [16] Williamson D F,Parker R A, Kendrick J S. The box plot: A simple visual method to interpret data[J]. Annals of Internal Medicine, 1989,110(11):916-921.
- [17] Kampstra P. Beanplot: A boxplot alternative for visual com-parison of distributions[J]. Journal of Statistical Software,2008,28(C01) :1-9.
- [18] Johnson T,Kwok I,Ng R T. Fast computation of 2-dimen-sional depth contours[C]// Proc of KDD'98,
- [19] Ester M, Kriegel H P, Sander J,et al. A density-based algo-rithm for discovering clusters in large spatial databases with noise[C]// Proc of KDD' 96, 1996:226-231.
- [20] l-Kiswany S, Ripeanu M. A software-defined storage for workflow applications[C]// Proc of 2016 IEEE International Conference on Cluster Computing (CLUSTER) , 2016: 350-353.

- [21] Breunig M M, Kriegel H P, Ng R T, et al. Optics-of: Identifying local outliers[C] // Proc of European Conference on Principles of Data Mining and Knowledge Discovery, 1999:262-270.
- [22] Breunig M M, Kriegel H P, Ng R T, et al. LOF :Identifying density-based local outliers[J]. ACM Sigmod Record, 2000, 29(2):93-104.
- [23] Tang J, Chen Z, Fu A W C, et al. Enhancing effectiveness of outlier detections for low density patterns[C]// Proc of Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2002: 535-548.
- [24] Radovanovic M, Nanopoulos A ,Ivanovic M. Reverse nearest neighbors in unsupervised distance-based outlier detection[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(5) :1369-1382.
- [25] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// Proc of KDD' 96, 1996:226-231.
- [26] Ng R T, Han J. CLARANS: A method for clustering objects for spatial data mining[J]. IEEE Transactions on Knowledge and Data Engineering ,2002, 14(5) :1003- 1016.
- [27] Karpis G, Han E H, Kumar V. Chameleon: Hierarchical clustering using dynamic modeling[J]. Computer, 1999. 32(8) :68-75.