CISC7107 Data Mining and Decision Support Systems

Assignment 01

Building an accurate and reliable classification model

Name: Yu Jincheng

Student Number:

Catalogue

1. Imbalance Datasets (Wine Quality Datasets)

1.1 Datasets Overview

The Datasets is retrieved from Kaggle, and the detail of the features is shown below (Table 1-1). This datasets is related to red variants of the Portuguese "Vinho Verde" wine. The wines are scored on a scale of 1 to 10(integer) based on 11 characteristics.

ID	Characteristic Name	Characteristic Description
1	Fixed acidity	Fixed acidity is the set of the wine's natural acids that we have already seen before (tartaric, malic, citric, succinic and lactic).
2	Volatile acidity	Volatile acidity (VA) is a measure of the wine's volatile (or gaseous) acids.
3	Citric acid	Citric acid is one of the less commonly found acid's in wine.
4	Residual sugar	Residual Sugar (or RS) is from natural grape sugars leftover in a wine after the alcoholic fermentation finishes.
5	Chlorides	The Chlorides of wine.
6	Free sulfur dioxide	Free sulfur dioxide is the portion of SO2 that is free in the wine.
7	Total sulfur dioxide	Total Sulfur Dioxide (TSO2) is the portion of SO2 that is free in the wine plus the portion that is bound to other chemicals.
8	Density	The density of wine.
9	PH	The PH of the wine. The pH value of wine is linked to its acidity
10	Sulphates	The sulphates of wine.
11	Alcohol	The alcohol of wine.
12	Quality	The quality score of the wine range from 1 to 10 (integral).

Table 1-1 The Characteristics of the Wine Quality Datasets

This dataset scores the sample wines based on characteristics. As shown in Figure 1-1, the distribution of scores ranges from 3 to 8, and a large number of samples are concentrated in the 5 to 7 score range, while the best and worst quality wine samples are 16 and 6, respectively. So the distribution of categories in this dataset is very unbalanced. In reality, the best and the worst are the most important indicators, and people tend to be interested in good wine and avoid bad wine. Therefore, in the process of data mining for this data set, it is very necessary to balance the data

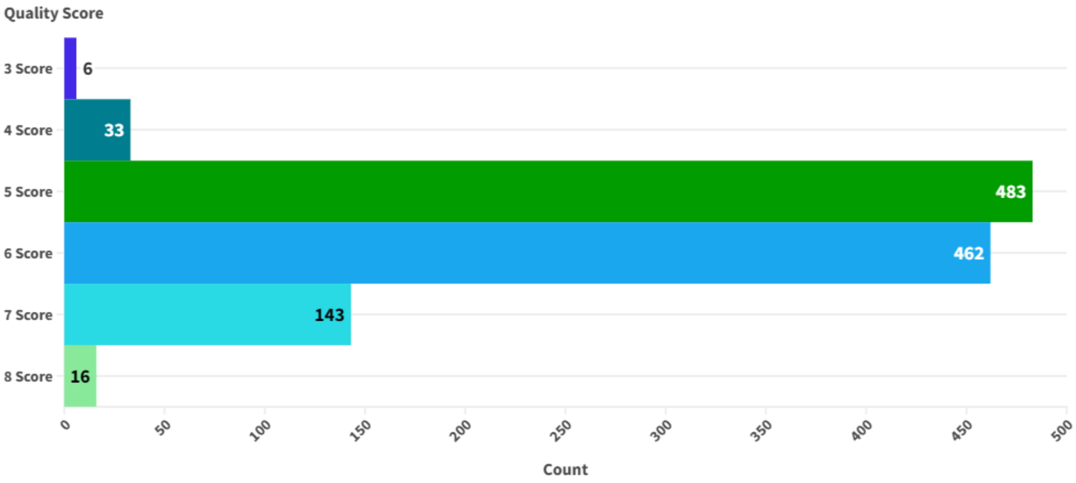


Figure 1-1 The diagram of the distribution of quality score

Figure 1-2 shows the data distribution of each feature and the distribution of different categories within each feature.

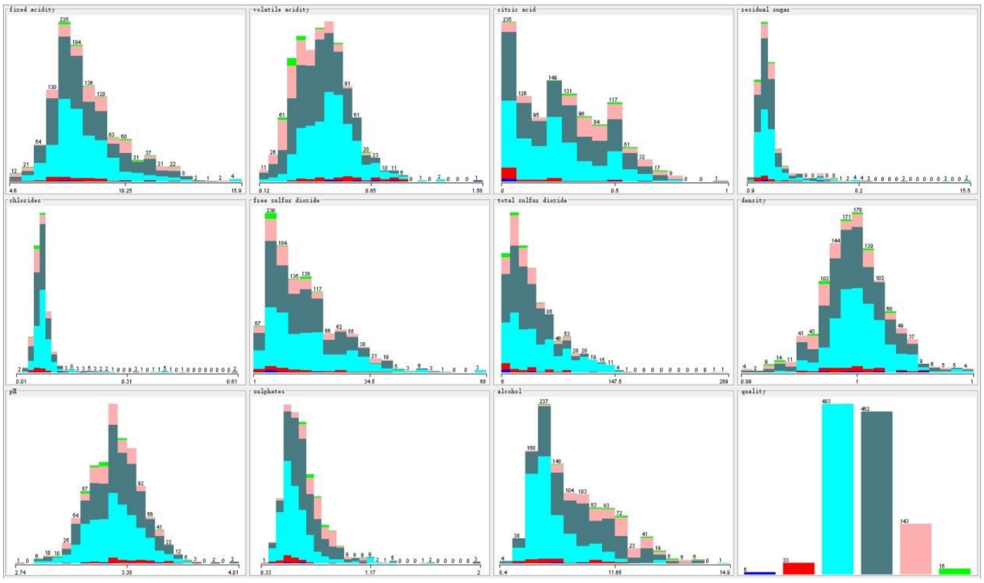


Figure 1-2 The diagram of the distribution of different classes within different features

1.2 Data Pre-processing

We used weka for data preprocessing. Our data preprocessing can be divided into three modules: data cleaning module, data balancing module and feature selection module. The specific weka algorithm used by each module is shown in Table 1-2. In the subsequent experiments, we will conduct ablation experiments on each module.

Module name	Method used (Weka)
Data Cleaning Module	Normalize + InterquartileRange + RemoveWithValues
Data Rebalancing Module	SMOTE + SpreadSubsample
Feature Selection Module	ChiSquaredAttributeEval

Table 1-2 Table of the Weka methods used in each our data pre-processing module.

In the data cleaning module, we first use weka's Normalize function to normalize the data and limit the range of data between 0 and 1. As can be seen from Figure 1-2, there are abnormal and extreme values in the data, so after data normalization, we use the InterquartileRange function in weka to screen outliers and remove the screened outliers with the RemoveWithValues function. The data distribution after data cleaning is shown in Figures 1-3. In the subsequent experiments, we verify the impact of adding and removing the data cleaning module on the classification results.

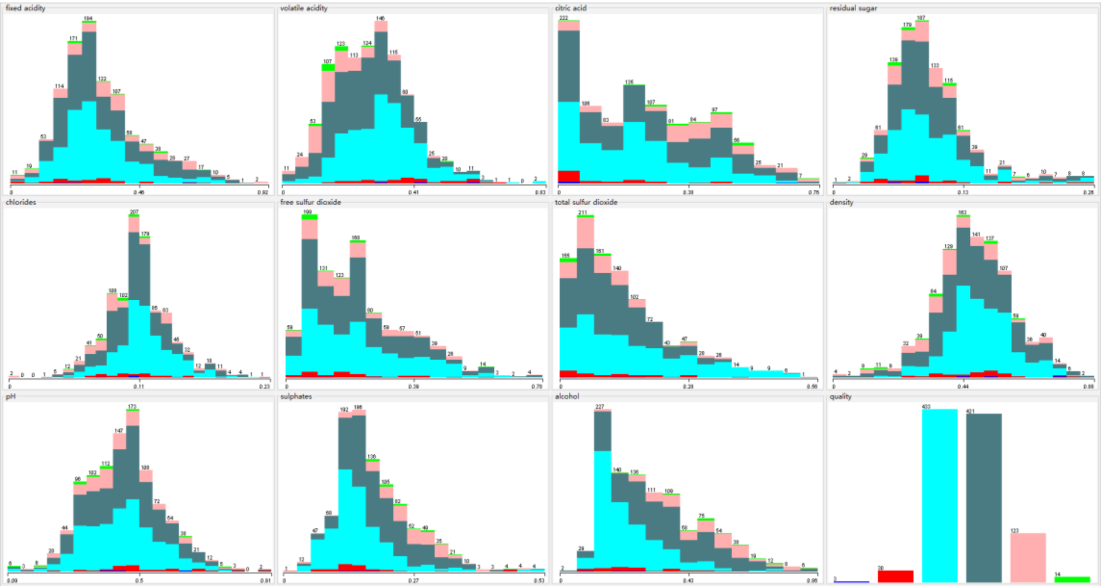


Figure 1-3 The diagram of the data distribution after cleaning the data.

Due to the significant imbalance of the original data, in the data balance module, we first use the SMOTE sampling algorithm in Weka to oversampling the data, and then use the SpreadSubsample function to randomly sample the oversampled data to obtain the balanced data of each category. The balanced data are shown in Figures 1-4, where the quantity distribution is consistent for each classification.

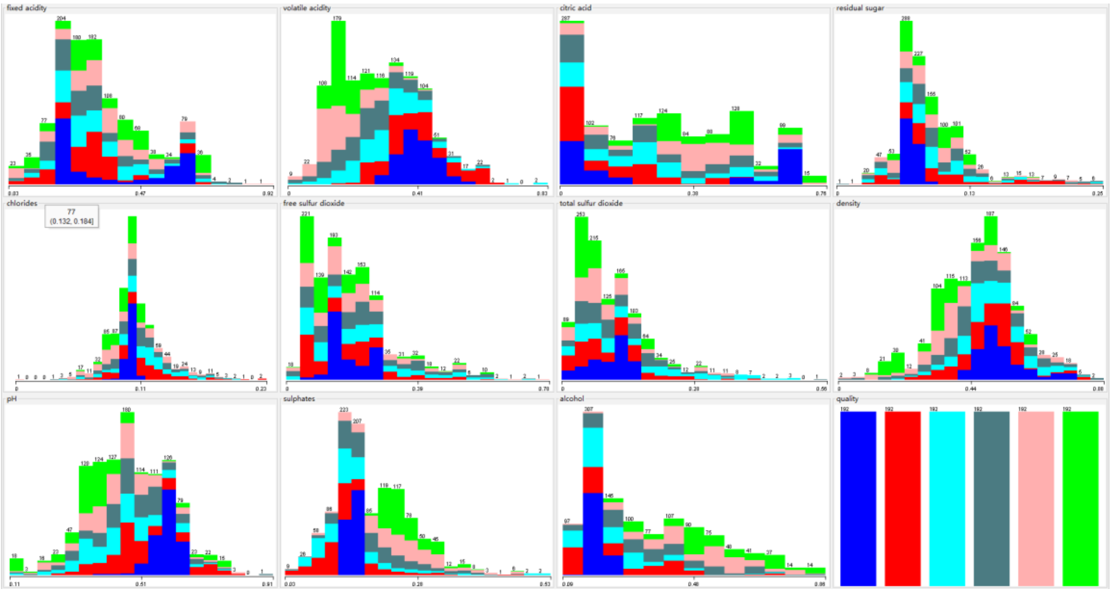


Figure 1-4 The diagram of the data distribution after data rebalancing.

In the data feature selection module, we used weka's ChiSquaredAttributeEval function to remove functions with low relevance to classification. In the experimental section, we verified the impact of the order of data balancing and feature selection on classification results.

1.3 Experiment

As shown in Table1-3(a), we divide the experimental data into seven groups, CBS, CSB, BS, SB, CB, B and O, according to the different pre-processing methods of the original data. Five classification algorithms in weka, J48, SVM, MLP, Random Forest and Naive Bayes, were used to model the classification of each group of experimental data. The effect of the classification model was then calculated using 10-fold cross validation. Finally, we conducted ablation experiment analysis on the classification results through the groups divided in Table 1-3(b) to explore the following four questions:

(1) For this data set, whether data cleaning can improve the accuracy of classification.

(2) For this data set, whether feature screening can improve the accuracy of classification.

(3) For this data set, whether the order of data balancing and feature selection affects the accuracy of the final classification results.

(4) For this data set, which is the optimal classification scheme among all experimental results?

Experimental Groups	**Experiment Method**
Clean+Balance+Select (CBS)	Cleaning + Rebalance + Feature Selection + J48
	Cleaning + Rebalance + Feature Selection + SVM
	Cleaning + Rebalance + Feature Selection + MLP
	Cleaning + Rebalance + Feature Selection + Random Forest
	Cleaning + Rebalance + Feature Selection + Naive Bayes
Clean+Select+Balance (CSB)	Cleaning + Feature Selection + Rebalance + J48
	Cleaning + Feature Selection + Rebalance + SVM
	Cleaning + Feature Selection + Rebalance + MLP
	Cleaning + Feature Selection + Rebalance + Random Forest
	Cleaning + Feature Selection + Rebalance + Naive Bayes
Balance+Select (BS)	Rebalance + Feature Selection + J48
	Rebalance + Feature Selection + SVM
	Rebalance + Feature Selection + MLP
	Rebalance + Feature Selection + Random Forest
	Rebalance + Feature Selection + Naive Bayes
Select+Balance (SB)	Feature Selection + Rebalance + J48
	Feature Selection + Rebalance + SVM
	Feature Selection + Rebalance + MLP
	Feature Selection + Rebalance + Random Forest
	Feature Selection + Rebalance + Naive Bayes
Clean + Balance (CB)	Cleaning + Rebalance + J48
	Cleaning + Rebalance + SVM
	Cleaning + Rebalance + MLP
	Cleaning + Rebalance + Random Forest
	Cleaning + Rebalance + Naive Bayes

Balance (B)	Rebalance + J48 Rebalance + SVM Rebalance + MLP Rebalance + Random Forest Rebalance + Naive Bayes
Origin Data (O)	J48 SVM MLP Random Forest Naive Bayes

Table 1-3(a) The table of the experimental groups

We performed several sets of ablation experiments as follows, as shown in Table 1-3(b). Group1 and group2 are mainly to verify the influence of the order of data rebalance and attribution selection on the classification results. Group3, Group4 and group5 are mainly to verify the influence of data cleaning on the classification effect of the final result. Group6 and Group7 are to explore whether feature selection can improve the accuracy of classification. The final Group8 is a comparison with the original data.

Experimental Groups	**Group1**	**Group2**	**Group3**	**Group4**	**Group5**	**Group6**	**Group7**	**Group8**
Clean+Balance+Select (CBS)	✓		✓			✓		✓
Clean+Select+Balance (CSB)	✓			✓				✓
Balance+Select (BS)		✓	✓				✓	✓
Select+Balance (SB)		✓		✓				✓
Clean + Balance (CB)					✓	✓		✓
Balance (B)					✓		✓	✓
Origin Data (O)								✓

Table 1-3(b) The table of each ablation experiment groups

1.4 Results and Analysis

1.4.1 Result Overview

Table 1-4 is an overview of all experimental results. Accuracy, Kappa, Execute Time, Average ROC

AUC, TP Rate and FP Rate are selected as the main comparison parameters in the report. We found that the set of experiments with the highest accuracy was those in which the original data were only subjected to data rebalancing operations and classified by Random Forest. It is the best in Accuracy, Kappa, TP Rate and TF Rate. At the same time, the classification effect of Random Forest modeling after cleaning and rebalancing the original data is optimal in the Average ROC AUC index, and there is little gap with other optimal groups in other indicators. In terms of operation time, Naive Bayes is undoubtedly the most time-saving algorithm, while MLP takes the most time but is also within the acceptable range.

Experiment	**Accuracy (%)**	**Kappa**	**Execute Time (s)**	**Average ROC AUC**	**TP**	**FP**
CBS + J48	70.3993	0.6448	0.09	0.852	0.704	0.059
CBS + SVM	42.3611	0.3083	0.13	0.654	0.424	0.115
CBS + MLP	67.7083	0.6125	1.08	0.88	0.677	0.065
CBS + Random Forest	78.9931	0.7479	0.42	0.96	0.79	0.042
CBS + Naive Bayes	55.9896	0.4719	0.01	0.871	0.56	0.088
CSB + J48	69.2708	0.6313	0.07	0.833	0.693	0.061
CSB + SVM	44.0972	0.3292	0.11	0.665	0.441	0.112
CSB + MLP	63.4549	0.5615	0.93	0.877	0.635	0.073
🥉 CSB + Random Forest	79.2535	0.751	0.33	0.957	0.793	0.041
CSB + Naive Bayes	52.7778	0.4333	**0**	0.874	0.528	0.094
BS + J48	68.4028	0.6208	0.01	0.852	0.684	0.063
BS + SVM	68.1424	0.6177	0.06	0.809	0.681	0.064
BS + MLP	60.3299	0.524	0.75	0.864	0.603	0.079
BS + Random Forest	78.3854	0.7406	0.22	0.954	0.784	0.043
BS + Naive Bayes	51.3889	0.4167	**0**	0.857	0.514	0.097
SB + J48	69.0104	0.6281	0.01	0.843	0.69	0.062
SB + SVM	59.6354	0.5156	0.07	0.758	0.596	0.081
SB + MLP	57.8993	0.4948	0.8	0.86	0.579	0.084
SB + Random Forest	78.7326	0.7448	0.22	0.952	0.787	0.043
SB + Naive Bayes	54.1667	0.45	**0**	0.861	0.542	0.092
CB + J48	71.9618	0.6635	0.1	0.857	0.72	0.056
CB + SVM	44.5313	0.3344	0.13	0.667	0.445	0.111
CB + MLP	70.8333	0.65	1.24	0.897	0.708	0.058
🥈 CB + Random Forest	81.3368	0.776	0.43	**0.962**	0.813	0.037
CB + Naive Bayes	58.5938	0.5031	0.01	0.888	0.586	0.083
B + J48	69.9653	0.6396	0.02	0.848	0.7	0.06
B + SVM	71.3542	0.6563	0.07	0.828	0.714	0.057
B + MLP	66.5799	0.599	1.17	0.884	0.666	0.067

🥇B + Random Forest	**81.4236**	**0.7771**	0.28	0.961	**0.814**	**0.037**
B + Naive Bayes	53.9063	0.4469	**0**	0.861	0.539	0.092
J48	55.818	0.3139	0.02	0.689	0.558	0.24
SVM	57.1304	0.2849	0.12	0.639	0.571	0.294
MLP	57.7428	0.3146	1.16	0.742	0.577	0.264
Random Forest	67.979	0.4832	0.31	0.834	0.68	0.203
Naive Bayes	54.1557	0.2973	**0**	0.722	0.542	0.234

Table 1-4 The table of the result overview (🥇:The Best, 🥈: The Second, 🥉: The Third)

1.4.2 Experiment Group8 (Compared to the Original Data)

In this report, we compared the classification accuracy and Kappa of the original data with and without data preprocessing. As shown in Figure 1-5, data preprocessing is very necessary, and the data after data preprocessing is significantly better than the original data without data preprocessing in terms of classification accuracy and Kappa.

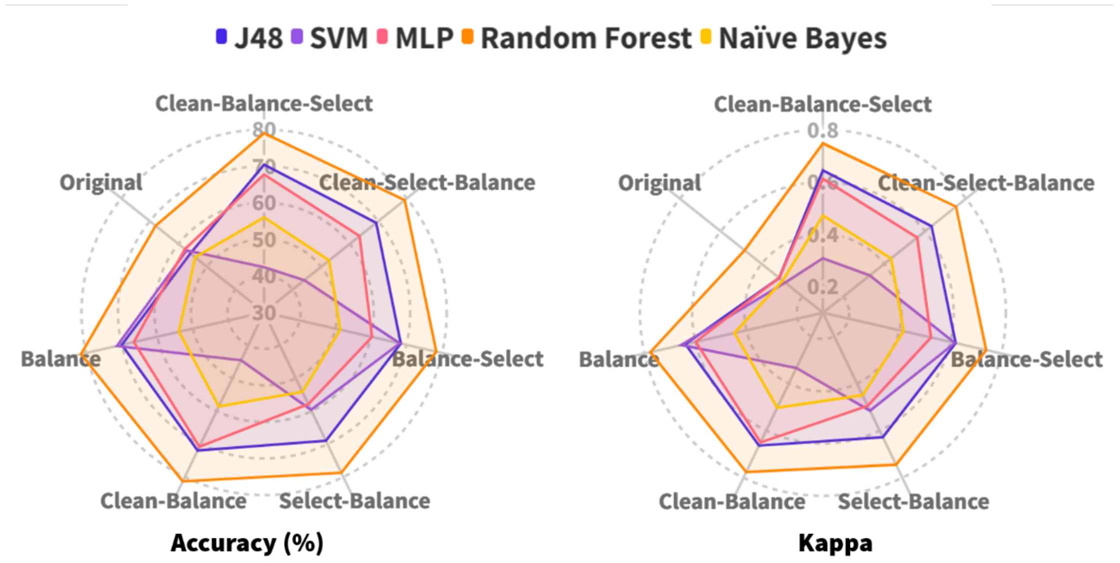


Figure 1-5 Compare the accuracy and kappa of data obtained through different preprocessing methods with the original data in various classification algorithms.

1.4.3 Ablation experiment Group1 and Group2 (The impact of the order of rebalancing and feature selection on the accuracy)

Figures 1-6 show the results of ablation experiments for data rebalance and feature selection sequence. We can find that the order of feature selection and data rebalancing has different effects on classification accuracy for different classification methods. Data rebalancing before attribute selection is helpful to improve the accuracy of MLP. However, attribute selection before data rebalancing is more conducive to the accuracy of data classification by Random Forest.

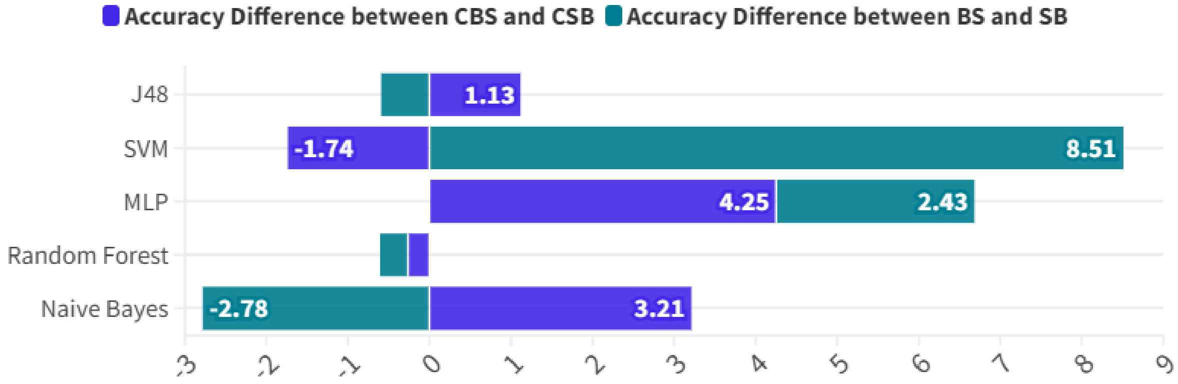


Figure 1-6 The figure of accuracy difference between CBS and CSB, and between BS and SB.

1.4.4 Ablation experiment Group3, Group4 and Group5 (The impact of data cleaning on accuracy)

According to Figure 1-7, we can find that data cleaning can improve the classification accuracy of J48, MLP, Random Forest and Naive Bayes. However, for SVM algorithm, data cleaning operation significantly reduces the accuracy of classification algorithm. We guess that SVM is not accurate enough to recognize the features, which leads to SVM not being able to distinguish these data well after data normalize operation.

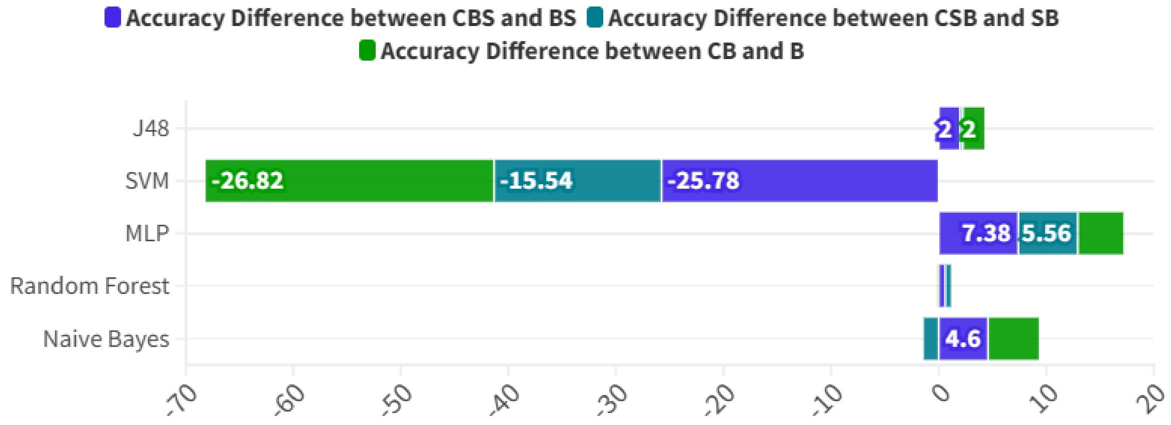


Figure 1-7 Ablation experiment Group3, Group4 and Group5

1.4.5 Ablation experiment Group6 and Group7 (Effect of Feature selection on classification accuracy)

According to Figures 1-8, we can find that the Feature Selection operation reduces the accuracy of all five classification algorithms with or without data cleaning. Among them, Feature Selection has the most significant effect on MLP. Through this ablation experiment, we speculate that because the number of features in this data set is not large, and the distribution of each feature is relatively independent, the classification accuracy may be reduced by feature screening.

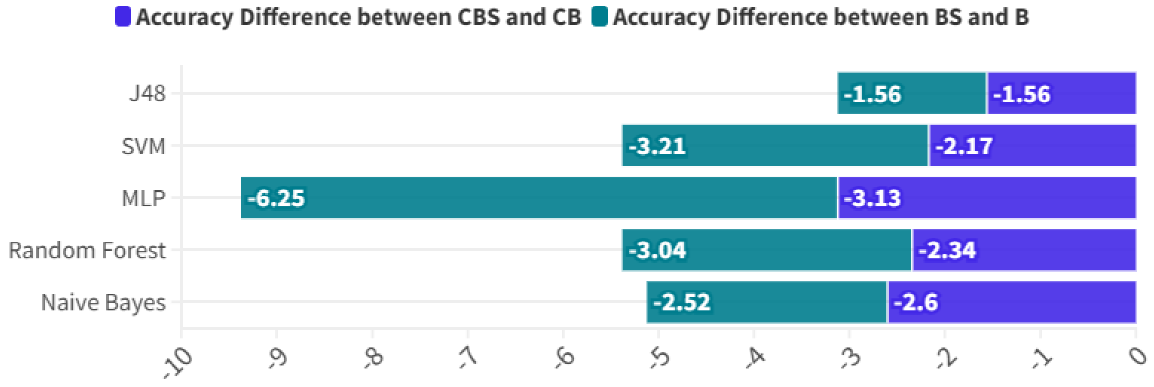


Figure 1-8 Ablation experiment Group6 and Group7

2. High Dimensionality Datasets (ShapeNetCore Datasets)

2.1 Datasets Overview

ShapeNetCore is a subset of the full ShapeNet dataset with a single clean 3D model and manually validated category and alignment annotations. It covers 55 common object categories with around 51,300 unique 3D models. In this paper, we select two kinds of 3D point cloud data in ShapeNetCore, Chair and Table, and classify the 3D point cloud data through various algorithms.

First of all, 3D point cloud data is of high dimension. In this data set, a point cloud data sample contains 2500 feature points, and each point contains x, y and z 3D coordinates, so the dimension of a data sample is 2500 * 3. As shown in Table 2-1, the training set contains 500 Chair 3D point cloud data and 500 Table 3D point cloud data. The test set contains 300 point cloud data for each category.

Class Name	Training Set Size	Testing Set Size
Chair	500	300
Table	500	300

Table 2-1 The table of training set size and testing set size

Table 2-2 shows several point cloud data samples, each 3D model is composed of 2500 points in 3D space.

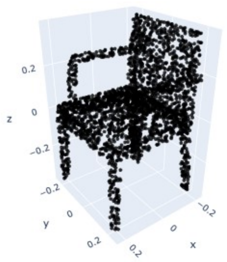
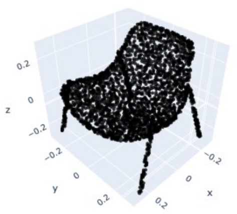
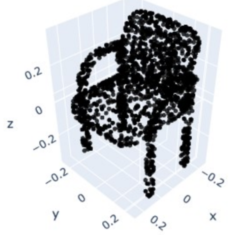
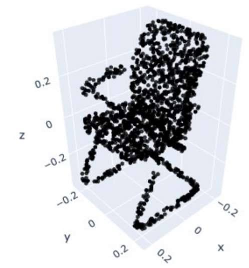
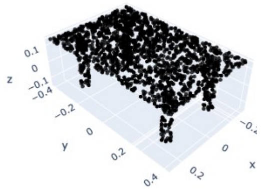
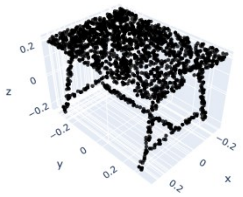
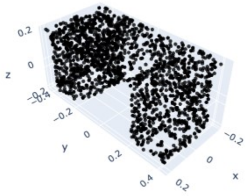
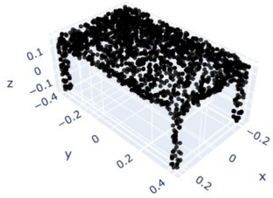
Class Name	Example 1	Example 2	Example 3	Example 4
Chair				
Table				

Table 2-2 The table of some example of the point cloud data for classification

2.2 Methods

2.2.1 Feature Selection method (OcTree)

Since each point cloud sample has 2500 points, we want to select these points. In this paper, we use OcTree to down-sample the point cloud data, and sample the point cloud point data from 2500 to 512,256,128,64 respectively. Figure 2-1(a) shows the Octree. Octree is a tree-shaped data structure used to describe three-dimensional space. Each node of the octree represents the volume element of a cube. Each node has eight child nodes. We pick any point in the node cube to represent all nodes in this cube.

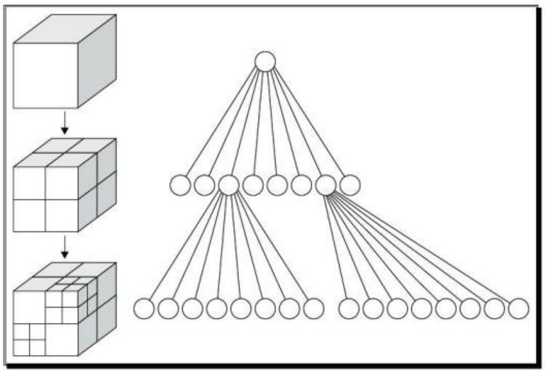
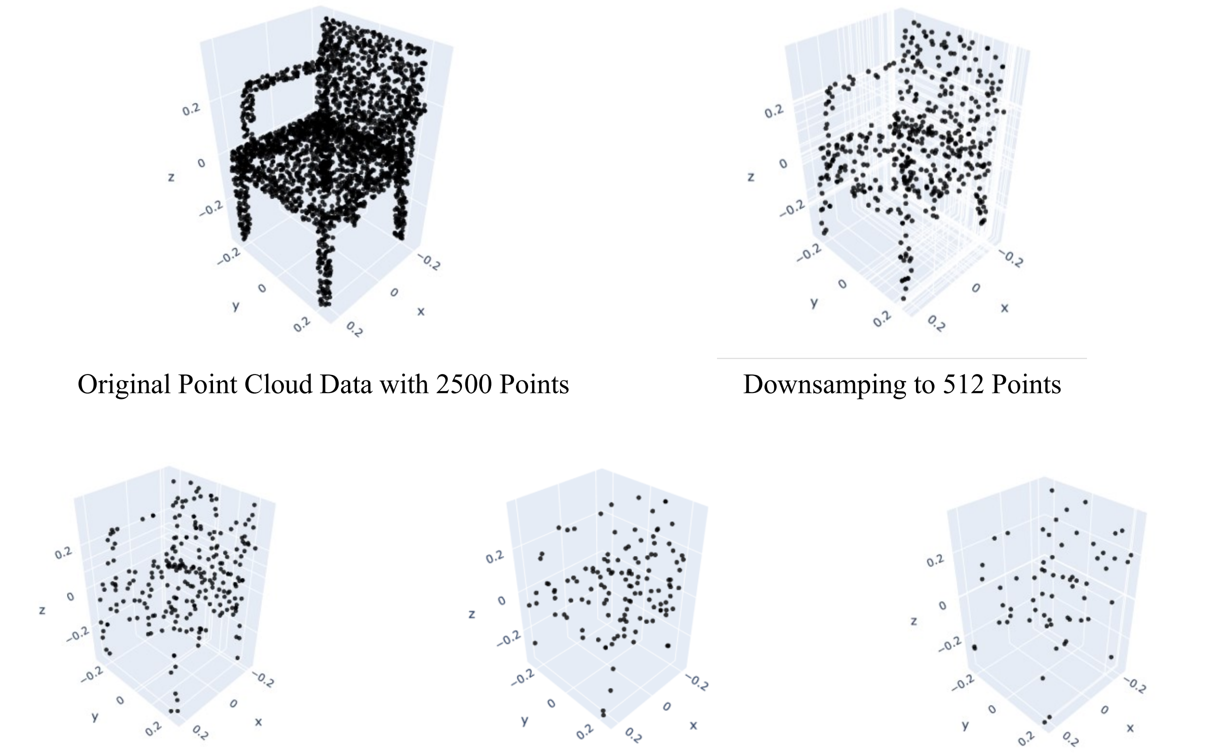


Figure 2-1(a) The diagram of OcTree

Figure 2-1(b) shows the effect of using OcTree to down-sample the point cloud at different sampling rates. We can see that the OcTree downsampling well preserves the spatial characteristics of the points in the point cloud, and removes the redundant points with similar spatial characteristics.



Original Point Cloud Data with 2500 Points

Downsamping to 512 Points

Downsamping to 256 Points

Downsamping to 128 Points

Downsamping to 64 Points

Figure 2-1(b) Situation diagram of point cloud data at different sampling rates

2.2.2 Traditional Classification Method

In this paper, we choose weka as one of the experimental platforms, and classify the point cloud data set by four classification algorithms: J48, SVM, Random Forest and Naive Bayes.

2.2.3 State of The Art Classificaation Method (PointNet)

In addition to using the classical classification Method in weka, we also tried to use python to build Pointnet classifier, a deep neural network-based Classificaation Method to classify our point cloud dataset. The main idea of PointNet is to treat the point cloud as an unordered set of points without considering the order or arrangement of points in the point cloud. It works by taking each point as input, extracting its local and global features, and learning to combine these features through a neural network for classification tasks. The core architecture of PointNet is a multi-layer perceptron (MLP), as shown in Figure 2-2(a), which is used to process the features of each point. For each point, it first processes its coordinates to obtain local features and then fuses them with global features. The local features are obtained by local coordinate transformation and local feature extractor, as shown in Figure 2-2(b), while the global features are obtained by aggregating the features of all points through the pooling operation.

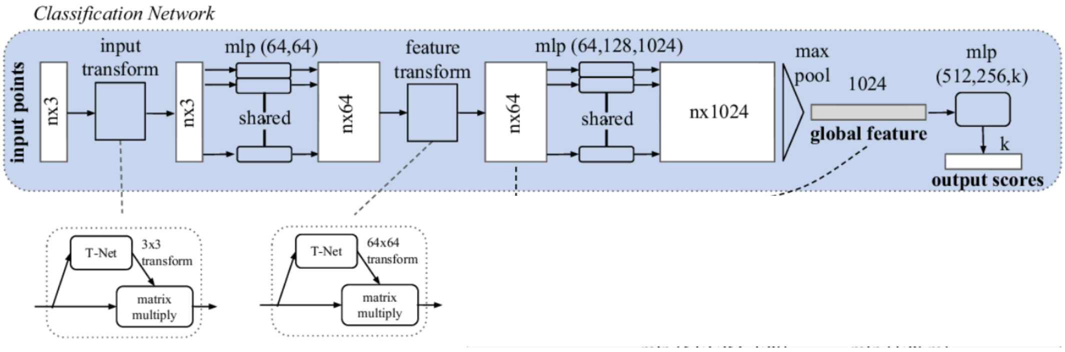


Figure 2-2(a) The figure of the classification Network of the PointNet

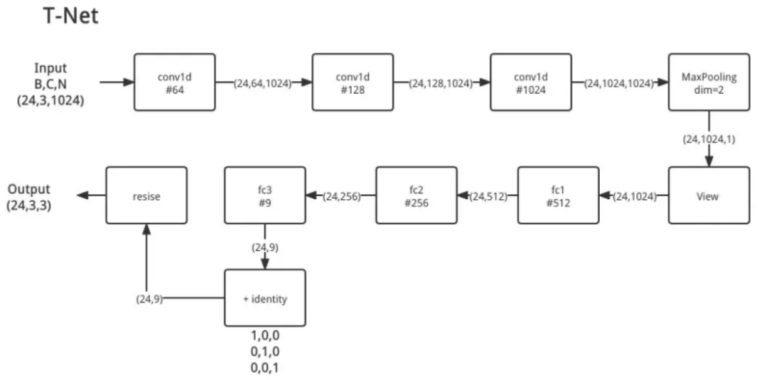


Figure 2-2(b) The figure of the T-Net for point cloud feature transform

2.3 Experiment Result

As shown in Table 2-3, the experiment with the best classification results is the one using PointNet to classify the point cloud dataset downsampling to 128 points. The classification accuracy was 95.723%, and the kappa index was 0.917. Secondly, Random Forest and Naive Bayes also have good results.

Experiment Name	Accuracy (%)	Kappa
Downsampling to 512 + J48	81.0	0.62
Downsampling to 512 + Naive Bayes	64.5	0.8651
Downsampling to 512 + Random Forest	94.8333	0.8967
Downsampling to 512 + SVM	88.3333	0.7667
Downsampling to 512 + PointNet	95.1	0.9010
Downsampling to 256 + J48	83.5	0.67
Downsampling to 256 + Naive Bayes	94.3333	0.8867
Downsampling to 256 + Random Forest	94.3333	0.8867
Downsampling to 256 + SVM	87.8333	0.7567
🥉Downsampling to 256 + PointNet	95.216	0.911
Downsampling to 128 + J48	79.5	0.59
Downsampling to 128 + Naive Bayes	94.3333	0.8867
Downsampling to 128 + Random Forest	95.3333	0.9067
Downsampling to 128 + SVM	87.8333	0.7567
🥇**Downsampling to 128 + PointNet**	**95.723**	**0.917**
Downsampling to 64 + J48	84.8333	0.6967
Downsampling to 64 + Naive Bayes	94.5	0.89
Downsampling to 64 + Random Forest	94.5	0.89
Downsampling to 64 + SVM	88	0.76
🥈Downsampling to 64 + PointNet	95.361	0.914

Table 2-3 The table of the result of the experiment in different classification method. (🥇:The Best, 🥈: The Second, 🥉: The Third)

3. Conclusion

3.1 Conclusion of Imbalance Datasets (Wine Quality Datasets)

We used J48(Decision Tree), SVM(Support Vector Machine), MLP(Multilayer Perceptron), Random Forest, Naive Bayes five classification algorithms to multi-classify the data after different data preprocessing. At the same time, we conducted ablation experiments for

different processes of preprocessing operations (data cleaning, feature selection, and data balancing).

We come to the following conclusions:

(1) The classification effect of the data after data preprocessing is significantly better than that of the original data without data preprocessing.

(2) Through our experiments, we conclude that the best solution for this dataset is to use random forest to classify the data with only data rebalancing. The accuracy of classification is 81.4236%, and the Kappa index is 0.7771 (10-fold cross validation).

(3) For all experimental groups, using random forest to classify this dataset is the best.

(4) It is helpful to improve the classification accuracy of MLP by performing data rebalancing before feature selection on this dataset. Feature selection followed by data rebalancing is more conducive to the accuracy of data classification by random forest.

(5) Data cleaning on this dataset can improve the classification accuracy of J48, MLP, random forest and Naive Bayes. However, it will significantly reduce the classification effect of SVM, which is a problem that needs to be explored in the future.

(6) The Feature Selection operation reduces the accuracy of all five classification algorithms with or without data cleaning operation. Among them, Feature Selection has the most significant effect on MLP. Through this ablation experiment, we speculate that since the number of features in this dataset is small and the distribution of each feature is relatively independent, the classification accuracy may be reduced by filtering the features.

3.2 Conclusion of High Dimensionality Datasets (ShapeNetCore Datasets)

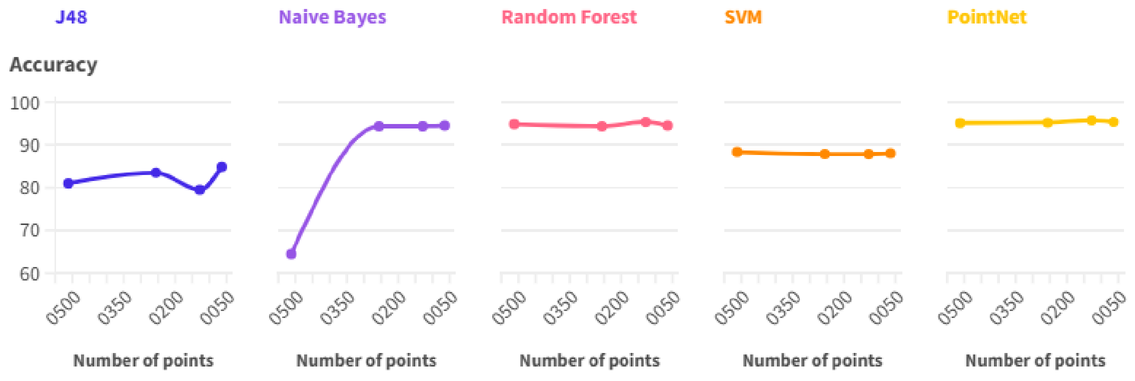


Figure 3-1 Plot of classification accuracy as a function of the number of points

Overall, the experiment with the best classification results is the one using PointNet to classify the point cloud dataset downsampling to 128 points. The classification accuracy was 95.723%, and the kappa index was 0.917. Secondly, Random Forest and Naive Bayes also have good results.

As shown in Figure 3-1, we can find that the classification accuracy of Random Forest, SVM and PointNet is not greatly affected by the number of sampling points. Both Random Forest and PointNet achieve optimal classification results when down-sampling to 128 points. However, the classification effect of J48 and Naive Bayes has a great influence on the number of sampling points. Among them, Naive Bayes performs very well when the sample points are less than 256.