Xi'an Jiaotong-Liverpool University
西交利物浦大学

Application of Machine Learning in the Classification of Students Grades

Shuchen Yuan (2036501)

Lab-B-Group-7

April 26, 2022

1 Introduction

The grade of students is a crucial part of the feedback of course design and teaching quality assessment. This research focuses on categorizing student by majors based on their corresponding marks for each question. The dataset provided in this project is a collection of student grade where more than 500 hundred students are involved. Those students mostly major in four specific programs which is labeled as “1”, “2”, “3” and “4” and students belongs to majors other than those 4 programs are regarded as ”0”. In both supervised learning and unsupervised learning, previous research has made progress on training classifiers and models on the classification of a similar dataset “IRIS”. Based on that, this experiment is required to implement 3 classifiers in supervised learning and 1 model in unsupervised learning after preprocessing the original data. K-neighbor-classifier [1], Naive Bayes classifier [2][3], and support-vector-machine [4] is applied in supervised learning as well as Kmeans [5] model is implemented to accomplish the unsupervised clustering based on Python. All the selected classifiers and models are optimized and evaluated in the end accordingly to the data feature and experiment result.

2 Data Observation

2.1 Basic data observation

Pandas library and Numpy library are the two most used libraries in Python. They provide powerful means of data preprocessing and transforming. In addition, Matplotlib is another library which is helpful when visualizing the data. In this experiment, through Pandas and Numpy, a simple observation on the provided dataset is made as well as visualizing the observation result through Matplotlib

After simply code implement, it is shown that the origin dataset is a collection of marks of student organized 515 rows and 7 columns and according to figure 1 the data collection is unbalanced perhaps due to the uncertainty of the performance during one exam and the limitation of the dataset structure. According to the definition, a data bias is regarded as factors that can influence the performance of classifiers and models. Therefore, this kind of unbalance is a data bias since classifiers are more likely to predict the most frequently appearing labels to obtain a higher accuracy. However, the program “0” is chosen to be deleted it. Because of the uncertainty number of majors in program “0” and it also can reduce the imbalance of the origin dataset.

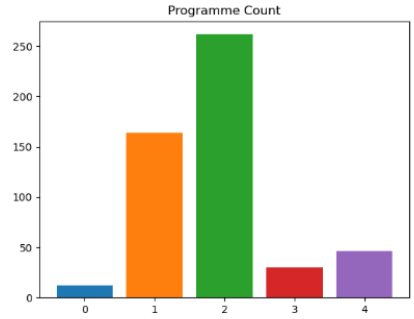

Figure 1. The distribution of students by program.

Then, looking through the dataset, the headmost and backmost 5 rows is taken to figure out the structure of it. As figure 2 reveals, there are still various outliers, missing values involved that requires data preprocessing. To start with, by using Pandas and Numpy, it is necessary to check and delete all the empty rows at the same time eliminating the duplicate of rows to reduce the similarity of the data by students. It is helpful for the following classification with more distinguishability.

ID	Q1	Q2	Q3	Q4	Q5	Programme
1.0	32.0	7.0	3.0	12.0	4.0	1.0
2.0	32.0	7.0	10.0	12.0	12.0	2.0
3.0	12.0	0.0	0.0	0.0	0.0	1.0
4.0	16.0	0.0	2.0	0.0	1.0	3.0
5.0	28.0	0.0	0.0	0.0	0.0	2.0

ID	Q1	Q2	Q3	Q4	Q5	Programme
511.0	34.0	5.0	10.0	20.0	20.0	2.0
512.0	14.0	7.0	10.0	2.0	0.0	1.0
513.0	22.0	1.0	10.0	0.0	6.0	0.0
514.0	24.0	0.0	10.0	2.0	4.0	0.0
NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure 2. Head and back most 5 rows of the dataset

As it is revealed, the given dataset concludes "ID" column to count the number of the students and "Programme" column is directly related to majors which ranges from 0 to 4. "Q1" to "Q5" column records the mark information of each question. Based on observation combining the actual situation, it is found the "ID" and "Programme" columns are the two irrelevant variables which are data biases for the classification then these 2 columns are deleted. In addition, to preserve as much information as possible about the original data, the first occurrence of the duplicate data is remained while other duplicates are deleted. In the end, after simply washing the data, the structure of the dataset is turned into 467 rows * 6 columns. Subsequently, the box figures of five programs and broken line graph of attributes of each question is plotted to further improve the data preprocessing as it is shown in figure 3 and figure 4.

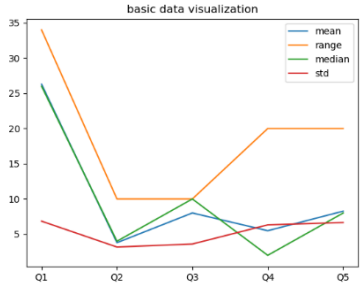

Figure 3. The graph of attributes of each question

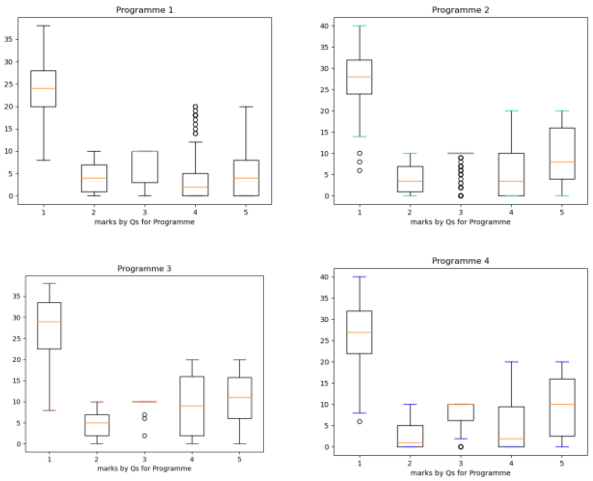

Figure 4. The box figures for each program

Generally, a box figure contains the median, range, upper quartile and lower quartile. It can reflect the skew of univariate distribution, average level of the data and the highlight the outliers which can be deemed as a data bias. In figure 3, the outliers are represented by a black cycle and median is the bar within the box. It is certain that overmuch outliers will restrict the performance of the models and the classifier. However, in reality, the outlier values are represented by minor probability events and is likely to happen. Additionally, it is clear that there are many outliers in Q3 and Q4 needed deletion but for small number of outliers in Q1 and Q2 is allowed. Following that, Q1, Q2 and Q5 are selected as candidate features instead of deleting the outliers. The reason is that deletion of large amount data will cause loss of the original information for a small-sized sample.

As followed, it is needed to dig more about the correlation of those three features and labels. Through roughly observation on figure 3 and figure 4, it is likely to discover that the mark codomain and distribution situation of Q1 is higher than other four questions. The appearing differences guide to explore more details about their relevance. Thus, the correlation matrix is given which displayed the correlation by the gradation of color.

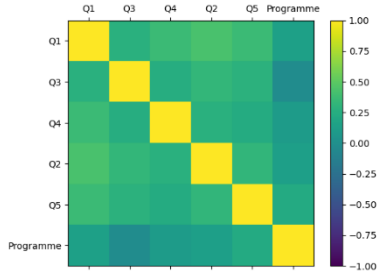

Figure 5. The correlation matrix of Qs and Programme

According to the drawn correlation matrix, it is blurred observed that the correlation score of different questions is around 0.25 The relevance between features and labels should be around 0.10. For more specific details the correlation thermal distribution diagram of the features and labels is drawn with Pearson coefficient which is defined as a linear correlation coefficient [6] that can directly evaluate the correlation between two set of data in figure 6.

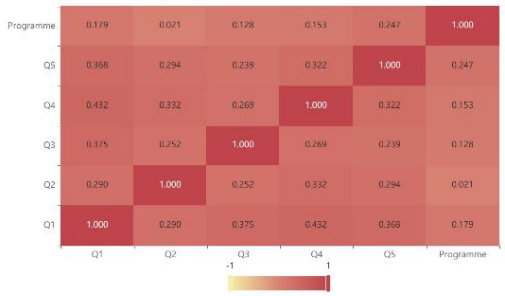

Figure 6. The Correlation thermal distribution diagram

In the correlation thermal distribution diagram, the value correlation coefficient is displayed with number which can directly infer the correlation of two data. Among candidate features Q2 has the lowest correlation rate which mean it is more likely to be distinguished. However, based on the explanation of the value of the correlation coefficient, the value in the range from 0.2 to 0.4 can all be defined as weak correlation. Thus, the result is acceptable for all candidate features even. In addition, Q5 is observed as the most related features to the "Programme" which is the label of the following machine learning. Therefore, after ensuring that the selected feature is not strongly correlated with other data, "Q5" is finally selected as a feature for next task.

2.2 Principal Component Analysis

Then, more complex mathematical approaches are applied to implement the data feature extraction. Firstly, Principal Component Analysis (PCA) [7] is taken as a method to implement dimensional reduction and feature extraction on the dataset.

Principal Component Analysis is based on the concept that transform overlapped dataset into new collection with less similarity between features. By calculating the covariance and covariance matrix of a given data to derive its eigenvectors, the mapping between different dimensions is completed to achieve the effect of dimensionality reduction. The most important step is to solve the eigenvectors of the covariance matrix. The corresponding formula for covariance and covariance matrix is below

$$\begin{aligned} Cov(X,Y) &= E[(X - E[X])(Y - E[Y])] \\ &= E[XY] - 2E[Y]E[X] + E[X]E[Y] \\ &= E[XY] - E[X]E[Y] \end{aligned}$$

Figure 7. Covariance formula

In the formula, E means the expectation of the given value. The value of covariance directly reveals the correlation of X and Y. Then, the value is arranged in given format to form a covariance matrix.

$$C = (c_{ij})_{n\times n} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix}$$

Figure 8. Covariance matrix

Finally, after basic mathematic operation, with the result of eigenvalues and eigenvectors of the covariance matrix, the original features are projected onto the K eigenvectors whose corresponding eigenvalue is the biggest K to obtain the new K-dimensional features. This procedure reduces the dimension at the same tine keep the integrity of the original dataset. In this experiment, we implement PCA algorithm with Python and visualize the result in 2 dimensions and 3 dimensions.

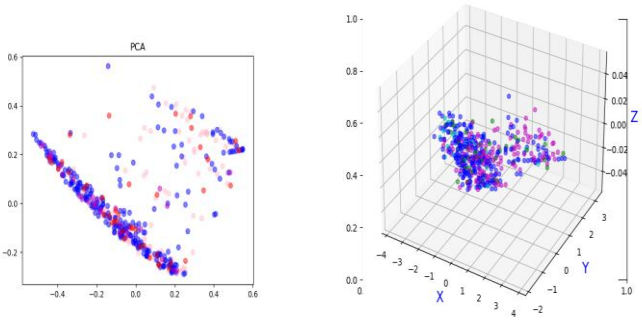

Figure 9. The PCA result

Based on the result, it is likely to observe that a part of data sample have been separated. Additionally, there are also few outliers lying in the upper area of the image and a large portion of the data is mixed together and the 3 dimensional PCA is mingled more seriously. However, it is able to blearily distinguish some of the dimension-reduced dataset is orthogonally distributing for which implies the new features have the maximum information retention of original features. Considering it still keeps information of the original data as well as significantly reduces dimensionality, the two-dimensional data of precessed by PCA is selected as 2 candidate features for supervised learning.

2.3 T-distributed stochastic neighbor embedding

Then, another approaches called t-SNE [8] is found more powerful for this task. The t-SNE algorithm based on the idea that the proximity of the distance distribution between data samples can be translated into magnitude of the probability. Based on the basic principle of SNE [9], in 2008, researchers improve the t-SNE by assuming the distribution models fit t-distribution which is superposition of t Gaussian distributions. Then it is found that t-SNE has advantages in gradient decline keep the two dissimilar point in a reasonable distance. It solves the crowding problem of the original SNE algorithm. In this experiment, the t-SNE model is implement by python codes and its visualization graph is as followed in figure 10.

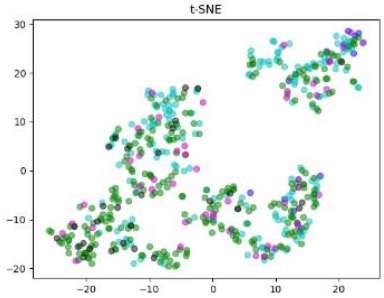

Figure 10. The t-SNE result

Accordingly, it is clearly revealed that the input data is clustered in five groups which is the number of the label "Programmme". Few outliers value is found in the graph. However, it can be predictable that four clusters are close and one is separated apart from those four. Based on the unbalanced structure of the dataset, the highest cluster might be classified by the influenced of students in major 0. In this way, due to its clear clustering and dimensionality reduction, the optimized two dimensions of t-SNE is selected as another two input features for supervised learning.

2.4 Brief Summary

To find databias, it is needed to find outliers value or factors that can weaken the performance of classifiers and models. By initially observing the dataset with visualization approach, it is found that the unbalance of sample distribution, occurrence of program 0 and the outlier values in scores of each question are the possible data bias which may restrict the performance

of machine learning models. Furtherly, the PCA and t-SNE result also show there are few outlier values and uneven clustering occurrence which probably related to unbalance revealed by figure 1. Taking into account the actual meaning of the sample and considering the maximum retention of the original features, the decision to remove the major 0 and keep the rest of the data deviations is made. After visualizing the correlation and evaluating the dimensionality reduction effect on the sample, "Q5", 2-dimension-PCA result and 2-dimension-t-SNE result as the selected features. Considering the possible loss carried out by dimensionality reduction, the original dataset is selected as a control group. Thus, there are nine finalized input feature which are divided into two collections and they will be compared and evaluated during the supervised learning in the following experiment.

3 Training Classifiers in Supervised Way

For supervised learning in this experiment, three commonly-used classifiers are implemented which are K-Neighbors-classifier [1], Naïve Bayes classifier [3] and Support Vector Machine [4]. In the following sections, brief introduction of the principles of the three classifiers as well as details about the optimization based on practical situation will be carried out. Based on the result, evaluation of the models and results will be given by selected evaluation metrics which is discussed in the next paragraph.

3.1 Selecting evaluation metrics

In this experiment, the proportion of results with correct training in the training set is deemed as the accuracy where the ratio of the test set to the training set is two to eight. According to a similar research based on WESAD dataset, the precision is not appropriate for an imbalanced dataset [10]. Thus, the harmonic average of precision and recall which is the f1 score is used where 'distance' is assigned to an inside parameter 'weights' considering the unbalanced distribution of data collections. For those classifiers who require hyper-parameter optimization such as KNN and SVM, the grid search is implemented in order to find optimal value for parameters by enumeration thus the accuracy calculated by grid search is used as the optimal result of the models. At the same time, K-fold cross validation is another commonly-used approach. It divides the dataset into K parts which has 1 set to train with K-1 set to test and can provide stable output for this experiment [10]. In addition, the values of standard division divided by mean of the cross validation score are used to reflect degree of dispersion of the result. To reduce the computational effort as well as obtain more information, the fold times is set to 5.

3.2 K-Neighbors-classifier

A. Basic Principle of KNN

The KNN model is provided by Sklearn library implemented by Python in this experiment. Based on minority-voting law, aiming to determine the category of the unknown sample, the model will calculate the distance between it and all known samples and select closest K sample to indicate the unknown sample classification [1]. Therefore, normalizing the data attaches significance in order to weaken the influence of the override of small values by large values.

B. Tuning the Model

In the KNN model, there is only 1 hyper-parameter, which is the number of the neighbors [10], that is mandatory to assign. In this experiment, similar to f1 score the string "distance" is assigned to "weight" parameter. Additionally, the Euclidean distance is selected as distance measurement. The following figure 11 shows the formula of Euclidean distance in n-dimension. It usually refers to the real distance between two points.

$$d(x,y) := \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_n-y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}.$$

Figure 11. The formula of Euclidean distance

Followed by using grid search mentioned previously, hyper-parameter optimization is operating on the number of the neighbors which is the K of KNN abbreviation. Furthermore, to ensure the reliability of the grid research result, the cross-validation curve with increasing K is drawn in a broken line graph in order to find the best K value of the number of the nearest neighbors.

```
The best parament of the number of neighbors
11
```

Figure 12. The grid search result.

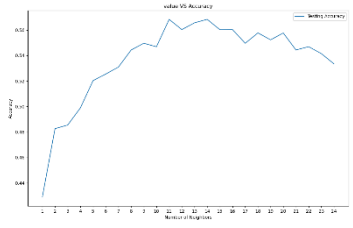

Figure 13. The cross-validation curve with increasing *K*

As it is displayed, grid search computes 11 is the best value for the hyper-parameter *n_neighbors*. Additionally,

the curve in figure 12 justifies the result is accurate because the peak of the curve occurs when K equals 11. Later, the final classification results and the corresponding accuracy rates are obtained after setting *n_neighbors* to 11. The output tables will be given whose input are the original and extracted dataset after selecting the evaluation metrics.

C. Result and Evaluation

	Program1	Program2	Program3	Program4
Original data	27	64	2	1
Extracted data	30	62	2	0

Figure 14. KNN classification result

	Accuracy	Grid search	F1_score	CV(Mean)	CV(Std/Mean)	Time(s)
Original data	0.5638	0.5683(K=13)	0.9118	0.5575	0.0359	1.3882
Extracted data	0.5531	0.5175(K=24)	0.5292	0.5032	0.1011	2.1213

Figure 15. The evaluation metrics corresponding result**.**

Overall, for the finalized result of classification, it demonstrates that the classifier tends to predict the largest number of labels to achieve high accuracy instead of in a comprehensive and reasonable manner. Associating with the distribution proportion of each major and compared with the result of original set, the result of the extracted dataset eliminates the value of program4, which contradict with the data distribution. According to the analysis of data observation section, one reason is assumed to be the imbalance of the dataset as well as the other possible factors may result in this is the dimensionality reduction is supposed to cause a certain number of losses of information and features. Due to the minor number of the major3 and major4, information belongs to those two labels may be missing in the downscaling process.

Additionally, the accuracy metric in figure 15 is analyzed. On the whole, the scores of evaluation metrics are varying from 40 to 50 which is acceptable and reasonable for a data set with heterogeneous and unevenly distributed samples. Likewise, it is clear that almost all evaluation metrics illustrate the classification model is more effective in classifying the original data than in classifying the extracted data. For the original set, its accuracy of this run approaches and best score calculated by grid search and the f1 score is remarkable. However, the accuracy of the extracted data is higher than the grid-searched value which is an unreliable result. Likewise, the value of the division of standard division divided by mean is around 10% and the consumption running time of extracted dataset is bigger than the original data. All those error and differences mention suggest the dimensionality reduction leads to a more discrete distribution of the data. For the classifier who learn by distance, discrete distribution increases the computational effort as well restrict behaviors of the classifier due to the increased instability.

3.3 Support Vector Machine

A. Basic Principle of SVM

Support vector machine is defined as a binary classifier. Based on finding the optimal hyper-planes to distinguish the two classes with the best margins from the support vector, it is very effective for classifying small to medium sized, nonlinear and high-dimensional [11].

B. Tuning the Model

Through previous research, it is discovered that the facing problem is a multi-classification issue which it is speculated to require an one-to-many approach is used in SVM. Therefore, the SVC classifier is applied where its parameter *decision_function_shape* is set as “ovr” to activate the multi-classification function. According to the prior experience, the Gaussian radial basis kernel function is used in this case where the parameter kernel values “rbf”. Different from KNN, SVC classifier has two hyper-parameters. One is the penalty coefficient *C* and the other one is the Kernel coefficient *gamma*. The value of *C* is directly related to the fitting state of the classifier and *gamma* affects the computational power and performance of the kernel function. Similarly, the grid search is applied to implement hyper-parameter optimization along with the visualization of accuracy curve of the increasing *C* to testify the result. The optimized hyper-parameters are given in the following tables.

	The best parameter
C	13
gamma	scale

Figure 16. The hyper-parameter optimization

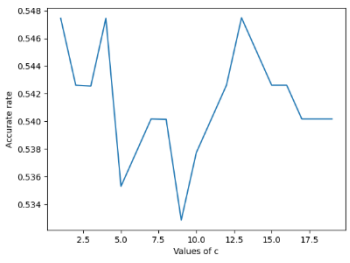

Figure 17. The curve of accuracy for increasing *C*

It is clear that the optimal parameters are obtained by C values 13 as well as the broken line graph testify the highest accuracy mark occur when *C* equals 13. In addition, the optimal value of *gamma* is the string "scale". Then, the

optimal SVM model used in this experiment is found as well as its classification result is demonstrated below.

C. Result and Evaluation

	Accuracy	Grid search	F1_score	CV(Mean)	CV(Std/Mean)	Time(s)
Original data	0.5631	0.5474(C=13)	0.5421	0.5425	0.0333	3.089
Extracted data	0.5425	0.5709(C=96)	0.4552	0.5602	0.02	0.7285

Figure 18. The result table of optimized SVM

	1	2	3	4
Original data	27	76	0	0
Extracted data	0	84	10	0

Figure 19. The evaluation metrics and corresponding result.

According to the figures above, for both data collections, the results exhibit the deficiencies of the dataset. The table in figure 18 show that the classifier is more likely to predict the most appearing label where the program 2 is given most frequently. However, the differences between the result of original data and extracted data reveal that the extracted dataset loses information of the original data for which in this case the classifier not follow the base distribution of dataset. It is supposed to predict the label as corresponding to their frequencies which roughly should be the number of program1 is larger program3. Additionally, comparing the result mention above with the result of KNN, SVC shows a more instability and a poorer performance of classification when it operates classification on the extracted dataset.

For the quantitative analysis, according to the accuracy and f1 scores, the original dataset outperforms the extracted dataset in the result of this run. The main reason is implied to be the loss by the dimensionality reduction. Namely, the difference of the running time between the two data collection testifies this because there is less information to process due to the loss mentioned above and the C value calculated by grid search is overlarge which indicates the occurrence of overfitting. However, it is noticeable that the accuracy result of the original dataset is higher than the mark of grid search which is supposed to be highest. The data bias of imbalance may be the reason causes this error which directly and furtherly justifies that the SVM classifier has an instable performance on such complicatedly-distributed data. The result needs to be improved so another classifier is required.

3.3 Naïve Bayes classifiers

A. Basic Principle of Naïve Bayes classifiers

The Naïve Bayes classifier [2] are functioning upon the Bayes rules which is an algorithm that describe the relationship between two independent conditional probabilities. The formula of it is given below.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Figure 20. The formula of Bayes rule.

The P (A) is generally defined as priori probability and P (A|B) is the posteriori probability. Naïve Bayes classifiers predict the corresponding distribution of each label by calculating the prior probability of the occurrence of them.

Among the Naïve Bayes classifiers branches there are a variety of models included. According to the previous observation, the datasets are more likely to distribute discretely thus the Categorical Naive Bayes classifier and the Multinomial Naive Bayes classifier are listed as candidate classifiers. The Complement Naive Bayes classifier is known for being suitable for dealing with unbalanced dataset for which it is selected due to the imbalanced distribution of the given data. After evaluating the selected two data collections in KNN and SVM, it is clear enough to figure out that models will be more effective in classifying on the original data instead of the extracted one. Therefore, the original dataset is applied in this case to elect classifier selected above and the evaluation of the optimal classifier will be provided as well.

B. Model Election

In this section, the optimal classifier is selected based on the stability of the model and the score of its training results. The input set is the processed original data and the scoring method is the same as mentioned metric previously except grid search because the parameter of Naïve Bayes classifier directly depends on the data collection. The result is as followed.

	Accuracy	F1_score	CV(Mean)	CV(Std/Mean)	Time(s)
CategoricalNB	0.5106	0.5414	0.5255	0.0600	0.0808
ComplementNB	0.4361	0.4861	0.5039	0.1047	0.0598
MultinomialNB	0.4255	0.4755	0.4690	0.0938	0.0593

Figure 21. The classification results of NB classifiers.

C. Result and Evaluation

The classification result is displayed from top to bottom according to the accuracy of the classifier. Accordingly, among the metrics that measure performance, the CategoricalNB classifier scores highest mark which indicates it the peak performance among those three models. For the stability, CategoricalNB models has the smallest value of the division of standard division divided by mean.

As mentioned before, the smaller value implies more concentrated data distribution which also indicate the sufficient stability of the CategoricalNB model in this case. In the end, the CategoricalNB model is selected due to the most accurate and stable characteristic.

3.4 Optimal models election

Then, result of the elected model CategoricalNB is compared with the previous result of KNN and SVM. The intuitive table is given below.

	Accuracy	F1_score	CV(Mean)	CV(Std/Mean)	Time(s)
KNN	0.5638	0.9118	0.5575	0.0359	1.3882
SVM	0.5631	0.5421	0.5425	0.0333	3.089
CategoricalNB	0.5106	0.5414	0.5255	0.0600	0.0808

Figure 22. The output of KNN, SVM and CategoricalNB

	Program1	Program2	Program3	Program4
KNN	27	64	2	1
SVM	27	76	0	0
CategoricalNB	44	44	6	0

Figure 23. Comparison between three classifiers.

Likewise, results are listed in ascending order of performance scoring for which it is patently that KNN reaches the highest score in plain accuracy, f1 scores and the cross validation in average. For stability evaluation, although the CategoricalNB runs by minimum processing time, the degree of dispersion of training results of it is twice as the other two classifiers which is assumed to be more unstable during classification. From the classification results, for SVM and CategoricalNB and the programs classification results do not follow the distribution pattern of the original dataset and the results of f1 scores also reflect well that the relationship between precision and recall cannot be well balanced. In the end, KNN model is selected as the optimal model for original data set due to the biggest performance scores and the stable output.

4 Unsupervised Classification

4.1 Introduction to Kmeans

For the unsupervised learning, based on the results of the supervised learning, it is assumed that a model that learns by distance may perform more efficiently. Thus, the Kmeans model is selected. Its working principle can be described as assigning sample points to the nearest, randomly selected cluster center by iteration and figuring out a new cluster center through calculating the distance to arrive at it until the location of the cluster center no longer changes. The distance is calculated using the Euclidean distance as the figure 11 shows. Finally, the model is built and results are obtained by calling the Kmean package in Sklearn library.

4.2 Unsupervised Learning Processes

For Kmeans, the only one mandatory parameter is the number of the clusters which is the meaning of K in Kmeans. Likewise, the selection process relies on visualizing the metric curve with increasing number of clusters. The measurement metric for this unsupervised learning is the sum of square due to error (SSE), normalized mutual information (NMI) and Silhouette Coefficient.

For a starter, the value of NMI can be seen as the extent to which one random variable contains the information of another random variable which can be interpreted as corresponding conditional probability. The peak of NMI result is implemented to indicate most eligible value for the parameter *K*. The result is as followed.

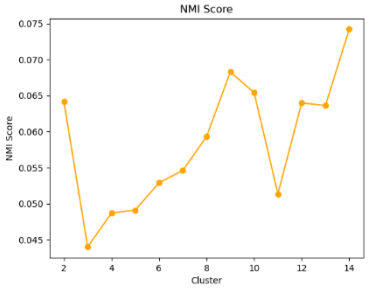

Figure 24. The NMI curve with increasing *K*

As the graph reveals, overall, the NMI index increases with increasing value of *K*. This is predictable that the highest score will occur when *K* is greater than fourteen. However, the truth is that *K* is related to the number of majors which seems impossible for the dataset according to the explanation of it. Thus, another measurement is the silhouette coefficient which calculated the ratio of intra-class distance to inter-class distance is used to measure the clustering scores whose result is given in figure 25.

Figure 25. The Silhouette Coefficient with increased K

A higher value of silhouette coefficient implies a more valid clustering result. As the graph displayed, the silhouette coefficient curve shows a opposite trend which the highest marks are at initial point. Namely, based on the previous research of the dataset, the number of the clusters is more reasonable to be greater than 3 hence there are at least four majors. Therefore, after comprehensive consideration, the number of K is selected to be 4. The reason that However, due to the contradiction between figure 24 and figure 25, the sum of square due to error is given to testify the result.

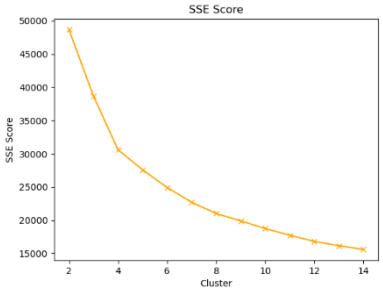

Figure 26. The SSE curve with increased K.

The SSE is used for measuring the shatter degrees and it is necessary to find the horizontal coordinate corresponding to the turning point of SSE graph. It is clear that the turning point occurs when K equals 4 which justifies the previous judgement. Therefore, the optimal value of the number of the cluster is 4. Then, the optimized Kmeans model is applied for clustering.

A. Clustering

Firstly, clustering and visualization of its result need to do the dimensionality reduction. From previous studies in the section 2.2 and 2.3 t-SNE has shown an intuitive clustering effect after dimension switching. Thus, the t-SNE is applied for dimensionality reduction before the clustering. In the end, the processed original data collection is input in the optimal Kmeans models and here is the result.

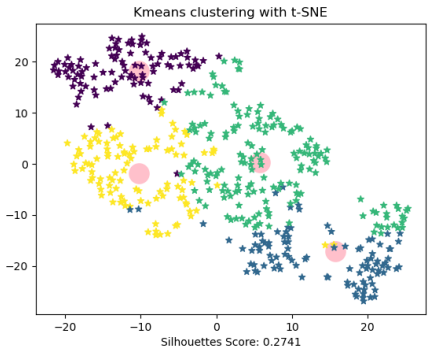

Figure 27. The clustering result of Kmeans.

4.3 Result evaluation

According to figure 27, clusters are separated and displayed by distinguishable colors. For quantitative analysis, the Silhouette Coefficient has a small value which indicates a mix of cluster may occur. The distribution of the clusters also justifies it because the cluster is not fully classified and the blue cluster are mixed with green cluster and yellow cluster. Moreover, the number of the cluster is four which is equal to the minimum number of the known programs of the dataset so it is possible to indicate the model is clustering according to those four specific majors and the program0 is not counted perhaps due to the minor size of it with few information to cluster. Overall, the Kmeans has a poor performance in this run.

5 Conclusion

In this experiment, the classification of student by majors is achieved through supervised learning and unsupervised learning. After the data bias is found and dealt with, a dimension-reduced and spliced dataset and the origin data are set as input for supervised learning to analyze the difference of the two data collections evaluate the performance of classifiers. Overall, the analysis of the results reveals that the unbalanced distribution of the dataset is a significant data bias affecting the effectiveness of the classifier, and further preprocessing such as oversampling and undersampling should be performed on the data while preserving the original features to the maximum extent. Likewise, all the models exhibit a certain degree of instability in the process of classifying the downscaled data, showing that the inevitable data loss in the downscaling process further causes data bias. Those pointed out limitation could be guidance and inspiration when processing the same or similar data sets in the future.

Reference:

[1] L. E. Peterson, “K-nearest neighbor,” Scholarpedia, vol. 4, no. 2, p. 1883, 2009

[2] K. P. Murphy et al., “Naive bayes classifiers,” University of British Columbia, vol. 18, no. 60, pp. 1–8, 2

[3] Naïve Bayes Classifier, [online] Available: https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[4] C. Cortes and V. Vapnik, “Support vector machine,” Machine learning, vol. 20, no. 3, pp. 273–297,

[5] G. Gan and M. Ng, "K-means clustering with outlier removal", Pattern Recognition Letters, vol. 90, pp. 8-14, 2017.

[6] J. Liu, Y. Zhang and Q. Zhao, "Adaptive ViBe Algorithm Based on Pearson Correlation Coefficient," 2019 Chinese Automation Congress (CAC), 2019, pp. 4885-4889, doi: 10.1109/CAC48633.2019.8997209.

[7] Moore and B., “Principal component analysis in linear systems: Controllability, observability, and model reduction,” IEEE Transactions on Automatic Control, vol. 26, no. 1, pp. 17–32, 1981.

[8] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.” Journal of machine learning research, vol. 9, no. 11, 2008.

[9] G. E. Hinton and S. Roweis, “Stochastic neighbor embedding,” in Advances in Neural Information Processing Systems, S. Becker, S. Thrun, and K. Obermayer, Eds., vol. 15. MIT Press, 2002. [Online]. Available: https://proceedings.neurips.cc/paper/2002/file/6150ccc6069bea6b5716254057a194ef Paper.pdf

[10] D. Bajpai and L. He, "Evaluating KNN Performance on WESAD Dataset," 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), 2020, pp. 60-62, doi: 10.1109/CICN49253.2020.9242568.

[11] S. Alam, Moonsoo Kang, Jae-Young Pyun and G. -R. Kwon, "Performance of classification based on PCA, linear SVM, and Multi-kernel SVM," 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), 2016, pp. 987-989, doi: 10.1109/ICUFN.2016.7536945.