
# Application of Artificial Intelligence based on student performance classification

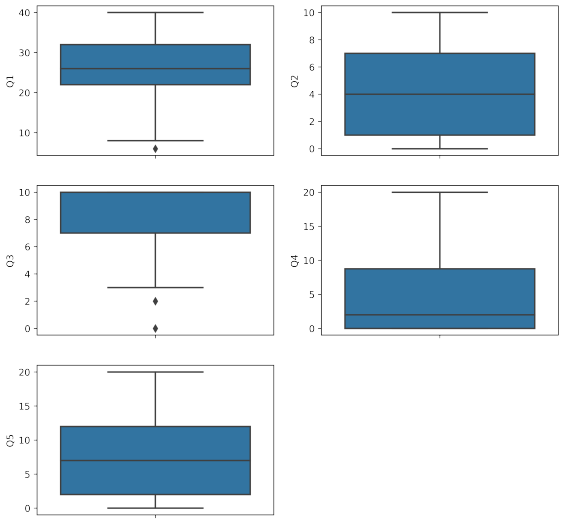
Shuchen Ji (2034172)

Lab-C-Group-5

May 23, 2022

# 1 Introduction

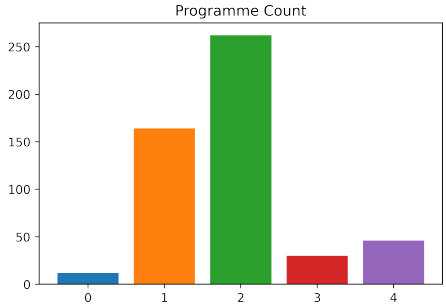
The coursework required this report for a spreadsheet of scores given to classify students into courses they belonged to based on the scores for each question and to show the full classification process. Three classifiers, vector machine, decision tree and random forest, were built for this experiment and the random forest was found to work best after analysis. Finally the clustering analysis was carried out by Kmeans.




# 2 Data Observation

## 2.1 Data checking and cleaning

First remove the row where the missing value is located using drop method of pandas to prevent the vacant value from having an effect on the experiment. Then use the bar chart to show the number of students per Programme:




The graph gives a small categorical sample of students in Programme 0 and they are not in the same major, so their scores have no categorical value and I simply remove them from the process.

In addition, for the information ID, the information entropy is too high as each student has a specific ID value. Especially for decision tree and random forest models, this can have a very negative impact. Therefore, the ID column is also deleted directly.

Use box plots to represent the scores for each question:

The box plots show that there are some outliers, but as some students score low and some score high, it is normal for outliers to exist. This report therefore considers these outliers to be valid information in relation to the student's professional classification and needs to be retained.

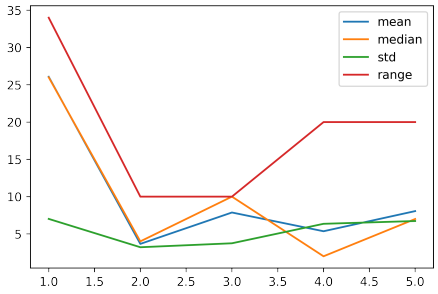
The final data obtained are:

| | Q1 | Q2 | Q3 | Q4 | Q5 | Programme |
|---|---|---|---|---|---|---|
| **0** | 32 | 7 | 3 | 12 | 4 | 1 |
| **1** | 32 | 7 | 10 | 12 | 12 | 2 |
| **2** | 12 | 0 | 0 | 0 | 0 | 1 |
| **3** | 16 | 0 | 2 | 0 | 1 | 3 |
| **4** | 28 | 0 | 0 | 0 | 0 | 2 |
| **...** | ... | ... | ... | ... | ... | ... |
| **507** | 32 | 0 | 10 | 2 | 0 | 1 |
| **508** | 30 | 1 | 10 | 5 | 0 | 2 |
| **509** | 26 | 0 | 7 | 0 | 4 | 2 |
| **510** | 34 | 5 | 10 | 20 | 20 | 2 |
| **511** | 14 | 7 | 10 | 2 | 0 | 1 |

502 rows × 6 columns

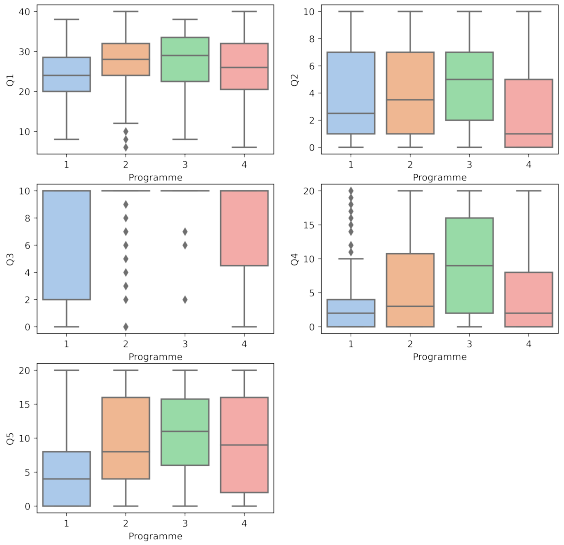
## 2.2 Data analysing

Through a line graph showing the mean, median, standard deviation and range of scores for each question.



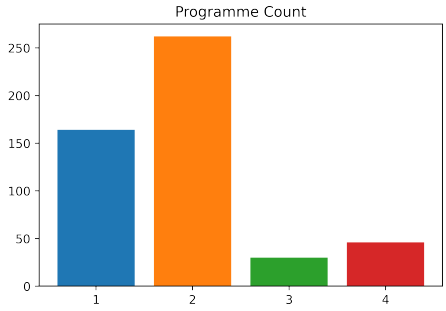

The distribution of mean and median was found to be relatively consistent for all questions and the distribution of standard deviation was smooth across questions. the higher mean score of Q1 may be due to the fact that Q1 is easier and therefore it is difficult to distinguish students by Q1 in later analyses. the range of Q1 is the largest and the range of Q2 and Q3 is the smallest, it is possible that in later analyses Q1 will perform worse as a feature in the clustering algorithm while Q2 and Q3 will perform better. The standard deviation was stable for all five questions.

Use Boxplot to indicate the performance of each profession on each question:




Students in Programme 2 and Programme 3 mostly scored higher despite some low scoring outliers on Q3, and it may be possible to tell Programme 2 and Programme 3 by Q3. Students in Programme 3 generally score higher on Q5. Students in Programme 4 generally scored lower on Q2. Students in Programme 1 performed less well on Q4 and Q5 compared to students in other Programmes, and I presume that Q4, Q5 may not be relevant to their major.

Use the bar chart to show the number of students per Programme:




The graph shows that there is a serious imbalance in the data. The number of students in Programme 1 and 2 is much higher than the number of students in the other Programmes, especially in Programme 2. This means that there are selection bias in the original sampling of the data. Therefore, this report uses Synthetic Minority Oversampling Technique(SMOTE) to remove this bias.

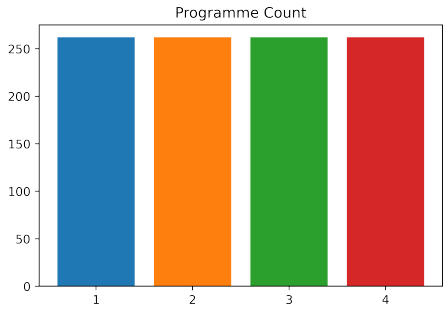
The basic idea of the SMOTE is to analyse a small number of samples and manually synthesise new samples to add to the data set based on a small number of samples[1].

The amount of data after oversampling is as follows:

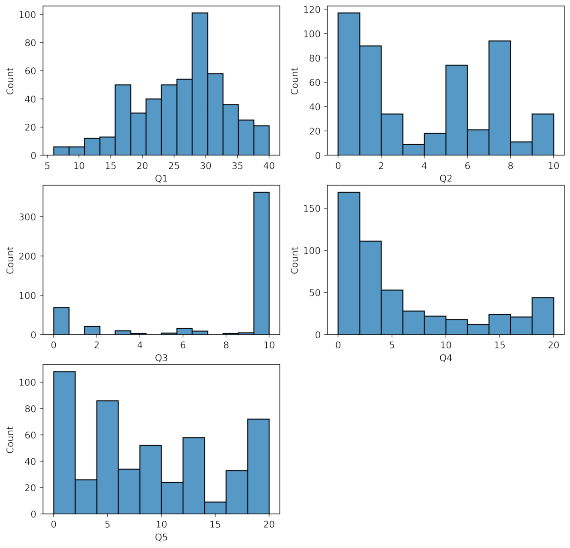
| | Q1 | Q2 | Q3 | Q4 | Q5 | Programme |
|---|---|---|---|---|---|---|
| **0** | 32 | 7 | 3 | 12 | 4 | 1 |
| **1** | 32 | 7 | 10 | 12 | 12 | 2 |
| **2** | 12 | 0 | 0 | 0 | 0 | 1 |
| **3** | 16 | 0 | 2 | 0 | 1 | 3 |
| **4** | 28 | 0 | 0 | 0 | 0 | 2 |
| **...** | ... | ... | ... | ... | ... | ... |
| **1043** | 32 | 0 | 8 | 3 | 2 | 4 |
| **1044** | 30 | 2 | 10 | 0 | 14 | 4 |
| **1045** | 12 | 0 | 1 | 0 | 0 | 4 |
| **1046** | 31 | 0 | 10 | 0 | 16 | 4 |
| **1047** | 18 | 1 | 9 | 0 | 2 | 4 |

1048 rows × 6 columns

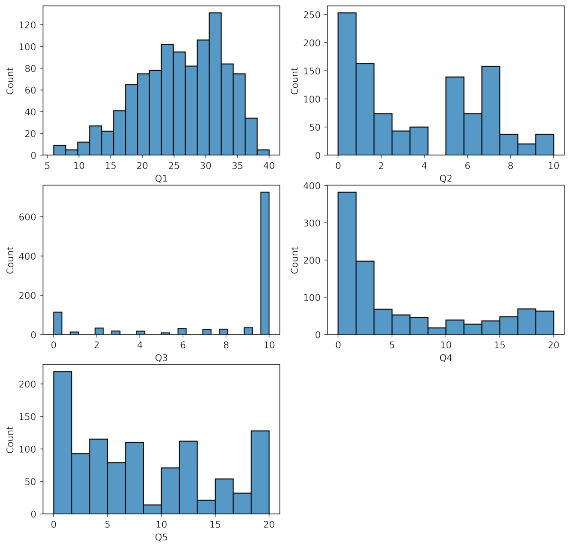
Use the bar chart to examine the number of students per Programme processed by SMOTE:



Draw a histogram for the distribution of the number of people in each score segment of each question.
Before using SMOTE:




After using SMOTE:

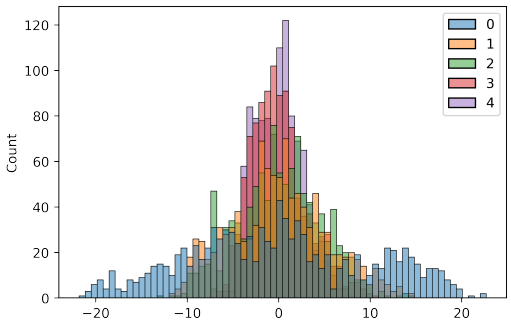



The distribution of the individual attributes did not change with the use of SMOTE, so my oversampling is justified.

## 2.3 Data feature extraction

### 2.3.1 PCA

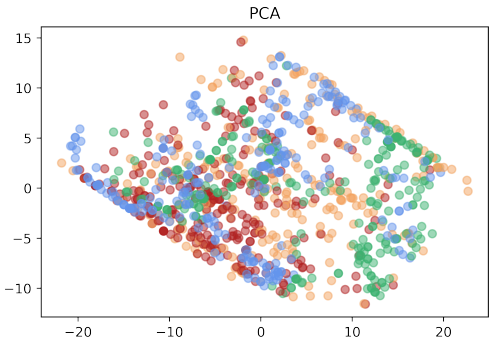
The Principal Component Analysis (PCA) algorithm maps high-dimensional data into a low-dimensional space by means of some linear projection, and expects the most information and variance in the projected dimension, thus using fewer dimensions of data while retaining the characteristics of more original data points[2].
The data features were extracted using PCA and the data distribution was obtained as follows:




It satisfies the normal distribution and its variance contribution of each feature is:

```
array([0.59403506, 0.1725379 , 0.13324744, 0.05892202, 0.04125758])
```

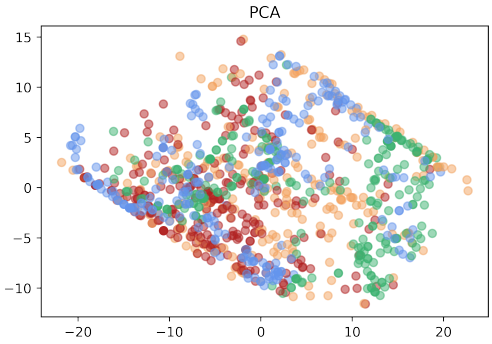
Each feature is similar to a normal distribution. PCA down to two dimensions:




The variance contribution of each feature is:

```
array([0.59403506, 0.1725379 ])
```

PCA down to three dimensions:




The variance contribution of each feature is:

```
array([0.59403506, 0.1725379 , 0.13324744])
```

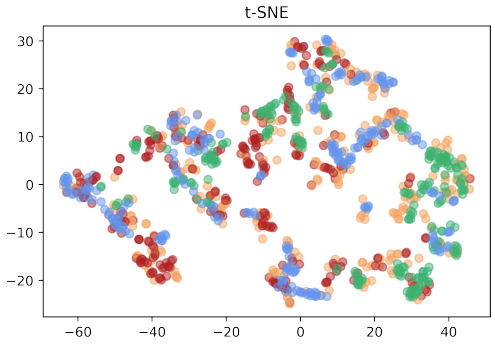
The first three variables were used to obtain nearly 90 percent of the information, so the first three features were used as the PCA features after dimensionality reduction.

### 2.3.2 TSNE

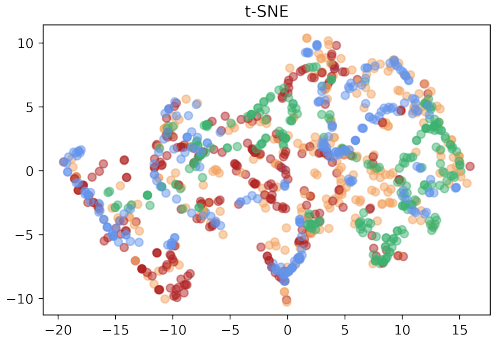
TSNE maps each data point to a corresponding probability distribution by means of a mapping transformation. Specifically, a Gaussian distribution is used to transform the distances into probability distributions in high-dimensional space, and a long-tailed distribution is used to transform the distances into probability distributions in low-dimensional space, so that the middle and low distances in the high-dimensional space can have a larger distance after the mapping, enabling the dimensionality reduction to avoid focusing too much on local features and ignoring global features[3].

This report performs a TSNE downscaling of the data by setting the init value to 'pca', and changing the value of random_state several times. The following figure shows the effect when random_state=0.

TSNE down to two dimensions:




TSNE down to three dimensions:

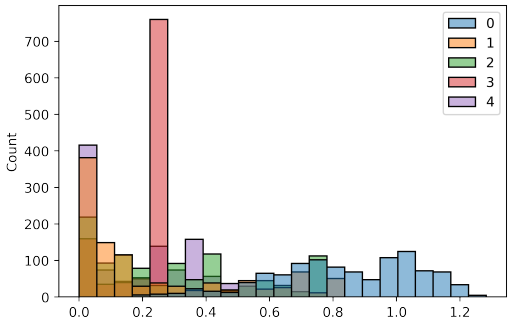



After replacing random_state for several times, the TSNE effect is still as shown above with a messy feature distribution, so the TSNE-processed features are not used for subsequent analysis in this report.

### 2.3.3 NMF

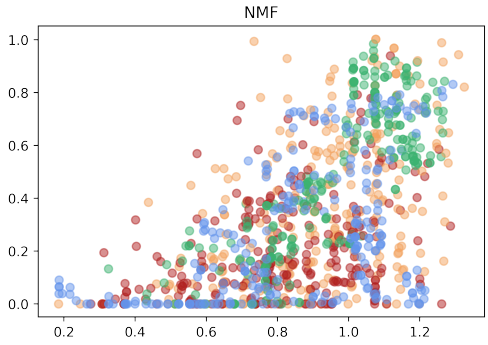
Non-negative matrix factorization (NMF), for any given non-negative matrix V, finds a non-negative matrix W and a non-negative matrix H that satisfy the condition V=W*H. Each column of the V matrix represents an observation and each row a feature; the W matrix is called the base matrix and the H matrix is called the coefficient matrix. NMF reduces the dimensionality of the original matrix by replacing it with the coefficient matrix H to obtain a reduced matrix of data features, thus reducing storage space[4].

The data features were extracted using NMF and the data distribution was obtained as follows:

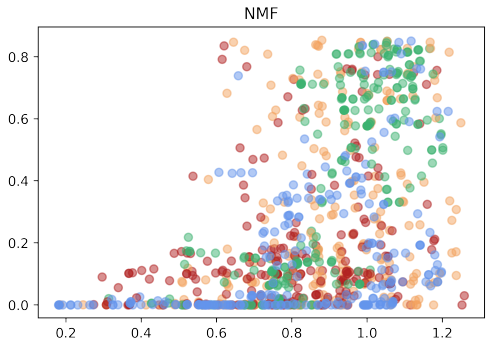



The figure shows that the distribution of the fourth and fifth columns is more abrupt, with some negative bias, which may allow the classifier to make wrong judgments.

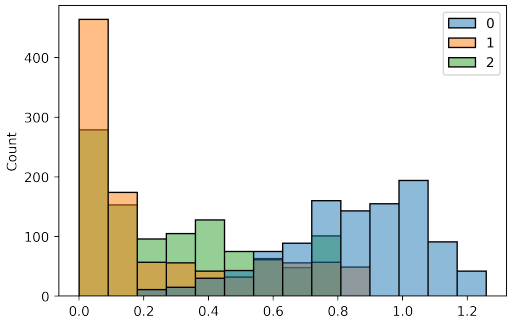
NMF down to two dimensions:




NMF down to three dimensions:




The graph clearly shows that dropping to 3D performs better, so only the first three columns have been retained for this report.
The final distribution obtained is as follows:




Therefore the first three features are used as the features after NMF dimensionality reduction.
In summary, the first three features of PCA and the first three features of NMF were selected as features. At this point, we have completed all the bias removal and feature extraction.

# 3 Training Classifiers in a Supervised Way

## 3.1 Data pre-processing

### 3.1.1 Merging data characteristics

Combine the 6 features selected in part2, and The data dimension is 1048 * 6.

### 3.1.2 Splitting the train set and test set

In Part 2 this report uses SMOTE to obtain new data. Here the larger volume of data processed by SMOTE is used as the training set and the original data not processed with SMOTE is used as the test set. The feature extraction step described above is also performed on the original data (the feature extractor used is the extractor trained from the oversampled data). The dimension of the test data obtained is 502 * 6.

### 3.1.3 Data standardisation

Because the data contains inevitable outliers, the report uses the Z-Score method to normalise the data.
Standardisation with zscore is based on the mean and standard deviation of the original data, it is intended to have a mean of 0 and a standard deviation of 1, which can place data of different magnitudes in the same matrix[5].
In order to fit realistic scenarios, a normaliser is first trained using the training data, which means train the mean and variance of the data, and this normaliser is then used to normalise the test data as well.

## 3.2 Classifier construction and classifier evaluation

This report uses support vector machines, decision trees, and random forests to perform the classification. When building the initial baseline model, a fixed random_state value is set to prevent the model from being built differently each time. The value used here is 42.
Due to the small data set, k-fold cross validation is used. It refers to dividing the original data into k groups, making a validation set for each subset data respectively, and taking the rest of the k-1 subset data as the training set to form K models. The average of the classification accuracy of the final validation sets of the K models is used as the performance index of the classifier under this K-CV[6]. On the training set, take the K value of each model as 3-10, and then take the average result of cross validation to evaluate the model.
The following results were obtained:

```
cv=3                    cv=7
DT_model score: 0.568   DT_model score: 0.592
RF_model score: 0.621   RF_model score: 0.651
SVM_model score: 0.445  SVM_model score: 0.460

cv=4                    cv=8
DT_model score: 0.584   DT_model score: 0.595
RF_model score: 0.644   RF_model score: 0.645
SVM_model score: 0.445  SVM_model score: 0.461

cv=5                    cv=9
DT_model score: 0.598   DT_model score: 0.581
RF_model score: 0.645   RF_model score: 0.638
SVM_model score: 0.450  SVM_model score: 0.463

cv=6                    cv=10
DT_model score: 0.594   DT_model score: 0.563
RF_model score: 0.646   RF_model score: 0.646
SVM_model score: 0.453  SVM_model score: 0.465
```

It can be clearly seen that the best model is the random forest model, regardless of the value of k.

## 3.3 Model testing

This report uses a random forest model trained on the training set and then taken to the test set for testing, and the following results were obtained:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1 | 0.96 | 0.96 | 0.96 | 164 |
| 2 | 0.99 | 0.96 | 0.97 | 262 |
| 3 | 0.94 | 1.00 | 0.97 | 30 |
| 4 | 0.90 | 0.98 | 0.94 | 46 |
| accuracy | | | 0.97 | 502 |
| macro avg | 0.95 | 0.98 | 0.96 | 502 |
| weighted avg | 0.97 | 0.97 | 0.97 | 502 |

View the confusion matrix:

```
[[159   3   0   2]
 [  6 252   2   2]
 [  0   0  30   0]
 [  2   0   0  44]]
```

It can be clearly seen that the recall rates for Programs 1, 2, 3 and 4 are all very high, and the data are well classified in the confusion matrix.

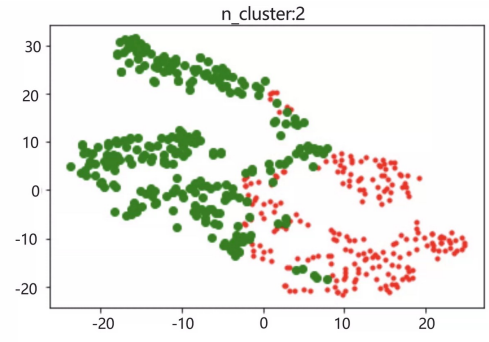
# 4 Unsupervised Classification

In this problem, the clustering analysis was carried out using the students' scores for each question rather than the previously extracted data features, as required by the question.

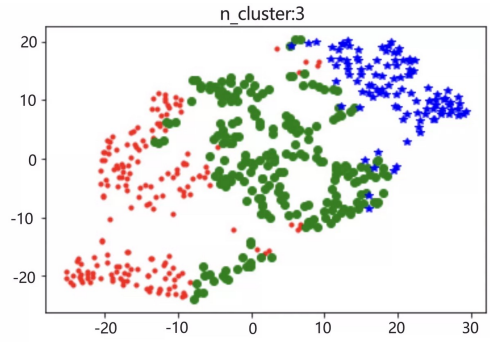
## 4.1 Kmeans

The main idea of Kmeans is to first randomly select k samples from the sample set as cluster centres, and calculate the distance between all samples and these k "cluster centres", and for each sample, divide it into the cluster with the nearest cluster centre, and for the new cluster calculate the new cluster centre of each cluster until the distance between the class cluster centre points The new cluster centroids are calculated for each new cluster until the change in cluster centroids is minimal or the specified number of iterations is reached[7].

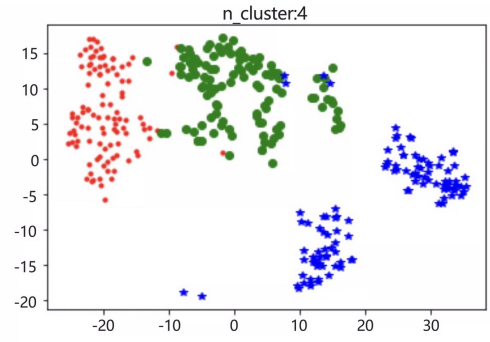
As this report is an unsupervised clustering, it should be assumed that the exact number of categories is not known. Therefore the number of clusters k is incremented from 2 and the clustering results are evaluated using homogeneity, completeness and silhouette coefficient and finally visualised. As the dimension of the data is greater than 2, it is necessary to use TSNE to reduce the dimension first and then visualise it.



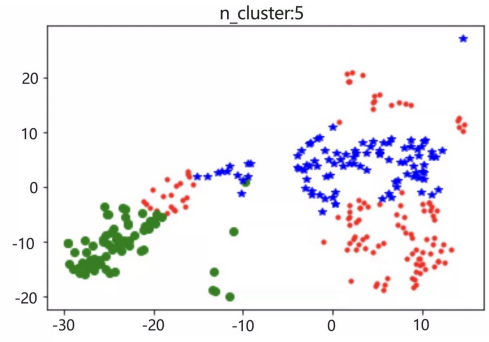

```
(0.0789542245120302, 0.06096714852359884)
0.3032288928417662
```




```
(0.042073378378862596, 0.04143718001786296)
0.26555336022670406
```




```
(0.040604303870303025, 0.04503840460385547)
0.2786307471784315
```




```
(0.03759451082960683, 0.04458266268243058)
0.26192750967625167
```

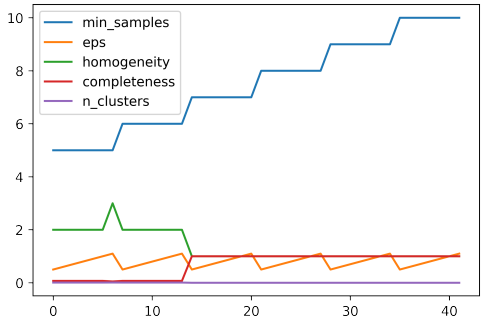
It can be seen that when the number of clusters is 2, the values of homogeneity, completeness and silhouette coefficient are the largest and the clustering effect is the best.

### 4.2 DBSCAN

DBSCAN defines a cluster as the largest set of densely connected points, enables regions with sufficient density to be classified as clusters, and allows clusters of arbitrary shape to be found in a noisy spatial database[8].

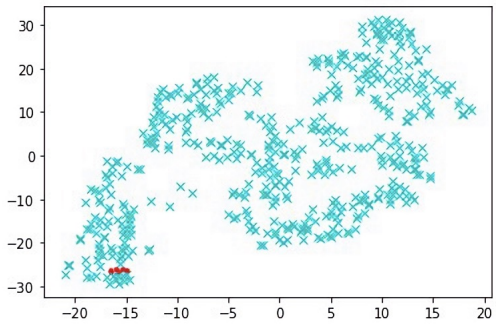
Because DBSCAN does not require us to specify the number of clusters, outliers are avoided and it works very well with clusters of arbitrary shape and size. That is why the DBSCAN clustering algorithm has also been tried in this report.

DBSCAN clustering has min_sample, eps and other parameters that need to be tuned. In this report, DBSCAN clusters are constructed and then manually tuned to observe the phenomenon. n_clusters for DBSCAN are calculated automatically, so no tuning is required. The tuning curves and tables obtained are as follows.




| | min_samples | eps | n_clusters | homogeneity | completeness |
|---|---|---|---|---|---|
| **0** | 5 | 0.5 | 2 | 0.071816 | 8.043088e-03 |
| **1** | 5 | 0.6 | 2 | 0.071816 | 8.043088e-03 |
| **2** | 5 | 0.7 | 2 | 0.071816 | 8.043088e-03 |
| **3** | 5 | 0.8 | 2 | 0.071816 | 8.043088e-03 |
| **4** | 5 | 0.9 | 2 | 0.071816 | 8.043088e-03 |
| **5** | 5 | 1.0 | 2 | 0.071816 | 8.043088e-03 |
| **6** | 5 | 1.1 | 3 | 0.050397 | 1.001433e-02 |
| **7** | 6 | 0.5 | 2 | 0.071816 | 8.043088e-03 |
| **8** | 6 | 0.6 | 2 | 0.071816 | 8.043088e-03 |
| **9** | 6 | 0.7 | 2 | 0.071816 | 8.043088e-03 |
| **10** | 6 | 0.8 | 2 | 0.071816 | 8.043088e-03 |
| **11** | 6 | 0.9 | 2 | 0.071816 | 8.043088e-03 |
| **12** | 6 | 1.0 | 2 | 0.071816 | 8.043088e-03 |
| **13** | 6 | 1.1 | 2 | 0.071816 | 8.043088e-03 |
| **14** | 7 | 0.5 | 1 | 1.000000 | 5.082368e-16 |
| **15** | 7 | 0.6 | 1 | 1.000000 | 5.082368e-16 |
| **16** | 7 | 0.7 | 1 | 1.000000 | 5.082368e-16 |
| **17** | 7 | 0.8 | 1 | 1.000000 | 5.082368e-16 |
| **18** | 7 | 0.9 | 1 | 1.000000 | 5.082368e-16 |

It can be seen that the model is optimal when the number of clusters is 2, eps is 0.5 and min_samples is 5 (disregarding the results when n_cluster is 1). To visualise the clustering results at this point:




As DBSCAN is an entirely density-based clustering method, and the data in this experiment did not satisfy this characteristic, poor visualisation results eventually emerged.

In summary, Kmeans clustering worked best for this data. When classifying students, it is possible to disregard their majors and directly classify them into two main categories for examination. For specific applications, the descriptive statistics within each cluster can be considered and the results can be combined with the students' majors to know which majors are most reasonably classified into one category.

## 5 Conclusion

This report first removes some bias that does not affect the experimental results by data checking and cleaning, and then removes selection bias from the data by SMOTE. Then the data were dimensionalised using PCA, TSNE and NMF, and the first three features of PCA and the first three features of NMF were selected as the results of feature extraction after analysis. Support vector machines, decision trees and random forests were then used to classify the data, concluding that the random forest model was better and using random forests for the analysis. Finally Kmeans clustering and DBSCAN clustering were carried out on the original data and it was concluded that Kmeans clustering was better, so that when classifying students, their majors could be ignored and they could be directly classified into two main categories for examination. In a specific application the descriptive statistics within each cluster can be considered and the results combined with the students' majors to know which majors are most reasonably classified into one category.

## References

[1] T. E. Tallo and A. Musdholifah, "The implementation of genetic algorithm in smote (synthetic minority oversampling technique) for handling imbalanced dataset prob-

lem," in *2018 4th International Conference on Science and Technology (ICST)*, 2018, pp. 1–4.

[2] A. Hadri, K. Chougdali, and R. Touahni, "Intrusion detection system using pca and fuzzy pca techniques," in *2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS)*, 2016, pp. 1–7.

[3] B. Dharamsotu, K. S. Rani, S. Abdul Moiz, and C. R. Rao, "k-nn sampling for visualization of dynamic data using lion-tsne," in *2019 IEEE 26th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, 2019, pp. 63–72.

[4] J. Tang, X. Ceng, and B. Peng, "New methods of data clustering and classification based on nmf," in *2011 International Conference on Business Computing and Global Informatization*, 2011, pp. 432–435.

[5] N. F. Abdullah, N. Rashid, K. A. Othman, and I. Musirin, "Vehicles classification using z-score and modelling neural network for forward scattering radar," in *2014 15th International Radar Symposium (IRS)*, 2014, pp. 1–4.

[6] D. R. S. Caon, A. Amehraye, J. Razik, G. Chollet, R. V. Andreão, and C. Mokbel, "Experiments on acoustic model supervised adaptation and evaluation by k-fold cross validation technique," in *2010 5th International Symposium On I/V Communications and Mobile Network*, 2010, pp. 1–4.

[7] I. El rube', "Image color reduction using progressive histogram quantization and kmeans clustering," in *2019 International Conference on Mechatronics, Remote Sensing, Information Systems and Industrial Information Technologies (ICMRSISIIT)*, vol. 1, 2019, pp. 1–5.

[8] S. Jebari, A. Smiti, and A. Louati, "Af-dbscan: An unsupervised automatic fuzzy clustering method based on dbscan approach," in *2019 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, 2019, pp. 1–6.