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

the Lab of Classifying and Clustering Based on Marks of Students from Different Majors

Yuzhen Chen (2036328)

Lab-B-Group-11

May 18, 2022

1. Introduction

1.1 requirements

In this lab, a spreadsheet containing students' marks for each question has been provided. There are three basic requirements of this lab. The first one is data observation, which aims at extracting data features from the supplied dataset and visualizing the extracted features. The second one is building classifiers, which requires designing appropriate classifiers using the extracted data features as the inputs. The third one is clustering, which needs to cluster students into groups due to the supplied mark of each question.

1.2 classifiers and the corresponding results

Four classifiers have been built in this lab which are Decision Tree, Random Forest, Logistic Regression as well as K-Nearest Neighbor respectively. None of the model accuracies can reach a relatively high level. The accuracies sustain 45 percent to 55 percent on average among the classifiers. Logistic Regression and K-Nearest Neighbor have a problem with data underfitting. Decision Tree and Random Forest has a problem with data overfitting. Among all the models Random Forest is recommended. The concrete recommendation reason and comprehensive elaboration are shown in part 3 of this report.

1.3 report structure

The rest of this lab report is organized as follows. Part 2 introduces data observation and focuses on feature extraction. Part 3 provides detailed evaluations of every classifier and the process of model training. Part 4 discusses classifying students into groups based on students' marks by one unsupervised learning algorithm. Part 5 concludes the report.

2. Data Observation

2.1 data description and data pre-processing

Data preprocessing and feature extraction are necessary before classifying and clustering. The basic data features such as range, mean value, median value and standard deviation have been visualized in this lab to acquire the general idea of data distribution and feature. The raw dataset is stored in a csv document which has 7 columns and 514 rows. The 7 columns record the information of students' IDs, Q1 marks, Q2 marks, Q3 marks, Q4 marks, Q5 marks and the programs students belong to. Every row in the spreadsheet records relevant data of each student and the students come from 4 different majors. The number of samples from majors 1 and 2 is far more than that from majors 3 and 4, which indicates the dataset is highly skewed. Data sampling is not used to eliminate the imbalance as the dataset is not rich. The dataset will shrink if sampling is used and the number of data will affect the performance of classifiers seriously. The csv document blends data of students from non-designated majors, which needs to be deleted. The classifying is based on marks and students' majors which indicates the ID column in the csv document can be considered an irrelevant variable that needs to be deleted. After basic data pre-processing, Q1, Q2, Q3, Q4 and Q5 are regarded as the original five independent features and the students' major is the data label.

2.2 basic data features

The average score, median score and standard deviation of Q1 to Q5 of each student are regarded as the data features called mean value, median value and standard deviation, respectively. The three data features have been visualized using matplotlib shown in figure1.1, figure1.2 and figure1.3, respectively.

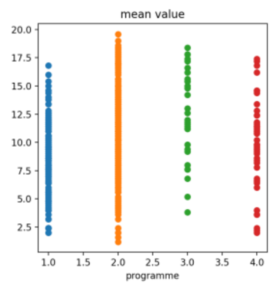

figure1.1

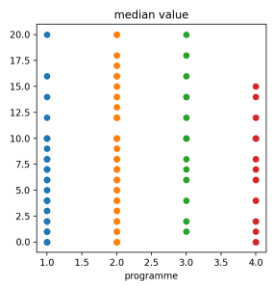

figure1.2

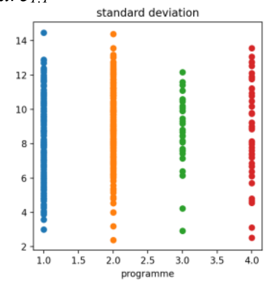

figure1.3

As the visualization results are shown in the figures generated by matplotlib, lacking significant data bias is the common problem among the three data features which may result in poor performance of classifying. Taking figure1.1 as an example, no significant data bias means that data points that have different labels have approximately the same distribution along the y-axis. In this lab, three complex mathematical operations have been used to attempt to generate data features capable of presenting significant data bias, which are PCA, T-SNE and NMF respectively.

2.3 PCA

PCA is the abbreviation of principal component analysis and it is useful to accomplish feature dimensionality reduction. The general mathematical process of PCA is as follows. Assume that a dataset consists of n-dimension features and needs to be reduced to k dimension. What PCA does is that finding k vectors which are linear independent denoted as u1, u2, …, uk onto which to project the data to minimize the projection error. In the case of this lab, the dataset is considered to have five features, therefore PCA can be used to reduce the dataset from five-dimension to two-dimension which is intuitive to visualize and show data bias. Extract the necessary data from the provided csv document to generate an N × 5 matrix as the input of PCA and the expected output is an N × 2 matrix. N is the number of sample data and 5 means the five features of the dataset. The visualization result of PCA by Python is shown in figure1.4. The different colors of the data points in the figure1.4 suggest distinct data labels.

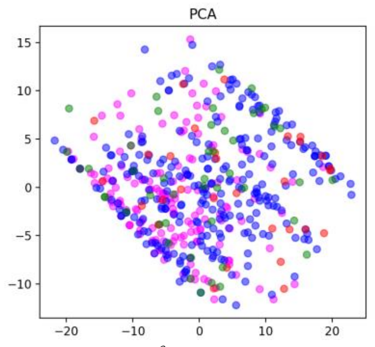

figure1.4

2.4 T-SNE

T-SNE is the abbreviation of t-distributed stochastic neighbor embedding and it is an efficient method of data visualization. The general idea of T-SNE comes from manifold learning. The basic principle of T-SNE is that each data point is mapped to the corresponding probability distribution by mapping transformation to realize dimensionality reduction. Taking the same input as the previous PCA and the visualization result of T-SNE generated by Python is shown in figure1.5.

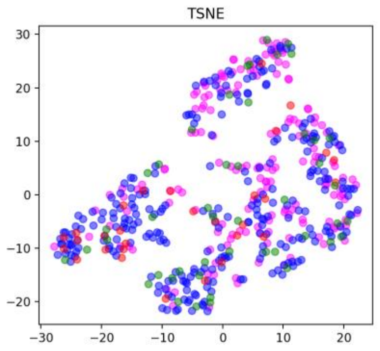

figure1.5

2.5 NMF

NMF is the abbreviation of non-negative matrix factorization. The general idea of NMF is as follows. Assume an arbitrary matrix V which is non-negative and V is an m×n matrix, m is the number of features and n is the number of samples. It can find two non-negative matrices W and H that satisfies the condition V= W×H. NMF is able to decompose one non-negative matrix into the product of two non-negative matrices. The matrix H is called the coefficient matrix. Replace the original input matrix with the coefficient matrix to reduce the dimensionality of features. In the case of this lab, NMF is used to reduce the dataset from five dimension to two dimension. Visualize the result of NMF by Python which is shown in figure1.6.

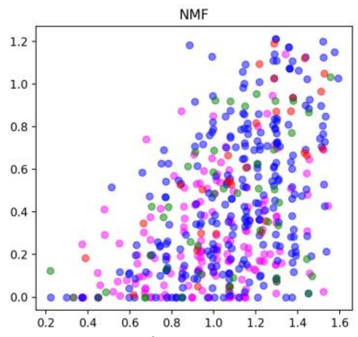

figure1.6

2.5 feature selection

The samples which have distinct data labels overlap with each other seriously among the visualization result of PCA, T-SNE and NMF. Intuitively on the grounds of the visualization results, none of the three mathematical operations help acquire beneficial and significant data bias. The provided raw dataset lacks data bias which results in the difficulty to find useful bias through some complicated mathematical calculations. Original data, PCA, TSNE and NMF are taken as an input of several classifiers for a quick test. The result suggests that the four inputs finally lead to similar accuracy rates and in certain cases taking the original data as an input performs more accurately. Finally, the original data is decided to be the input of the classifiers as data dimensionality causes partly information loss and the original data have similar or slightly more advantageous performance compared to data features that have been reduced dimensionality.

3. Building Classifiers in a Supervised Way

The classification is one algorithm type of supervised learning. Supervised learning algorithms use past data to predict the future [1]. The assignment of classification is that learn from the data from the training set so as to master the pattern of how they are classified. The correct prediction of classifiers implies putting it in the correct class when new data come [2]. The classifiers are essentially mathematical models which can learn from the previous data and classify new data. In this part, the experiment task is to build classifiers in a supervised way and evaluate the performance of the classifiers. Four different classifiers have been built in this lab which are Logistic Regression, K-Nearest Neighbor, Decision Tree and Random Forest.

3.1 experiment procedures

The experiment of designing and building each classifier conforms to the following research procedures. Firstly, preprocess the data and then randomly split the dataset into a training set and a test set in the proportion of 4:1. The dataset is not rich as there are only approximately 500 samples in the dataset so the proportion of 4:1 is comparatively proper. The training set is considered as the input to train the models and the test set is used to evaluate the performance of the models. The aim of the first step is to intuitively grab the performance of the classifiers. Secondly, debug the parameter to sort out the optimal parameters. Thirdly, the same method as the first step is to split the dataset and train the classifiers. Gain the prediction results of the classifiers with the input of the training set and test set respectively. 5-fold cross validation will be used to calculate the accuracy rates and the reason to apply 5-fold is subjected to the restriction of the small dataset. Calculate the recall value, precision value, f1-score and AUC value. The above-mentioned metrics contribute to evaluating the performance of the classifiers. The experiment of building classifiers is implemented by Python.

3.2 Logistic Regression

Logistic regression can deal with classification problems. The binary classification is taken as an example to illustrate the inference process of the classification of logistic regression. It exists only two classes in binary classification which is 0 class and 1 class. The core part of logistic regression is a trick function called sigmoid that is defined in the form of figure3.1.

$$h_\theta(x) = g(z) = \frac{1}{1+e^{-z}}$$

figure3.1

The sigmoid function has a special attribute, which is for any continuous input values and the output values have a range from 0 to 1. The output can be intuitively considered as the probability that g(z)=1 on the input z. A threshold can be set to 0.5. For example, if the output is larger than 0.5 then the data is classified in class 1 otherwise put it in class 0 [3]. The training process of this model is ingenious. In figure3.1 there is a part defined as $h_\theta(x) = g(z)$. The input z has a relation with the training data, which is shown in figure3.2.

$$z = \theta^T \cdot x$$

figure3.2

The formula is in matrix notation. θ^T and x in the formula are represented in vector form, T denotes the transpose of a matrix. θ^T is the parameter of variable x which is a matrix consisting of training data. The vital issue is how to calculate θ^T if the training data has been provided. The solution is to minimize the cost function of logistic regression, which is defined in the form of figure3.3.

$$Cost(h_\theta(x), y) = -y \cdot log(h_\theta(x)) - (1-y) \cdot log(1 - h_\theta(x))$$

figure3.3.

y in the formula is the corresponding data label of training data. Input the training data x and data label y then the optimal θ can be calculated by using some complicated mathematical methods such as gradient descent to minimize the cost function. The logistic regression is able to classify new data as the parameter θ has been obtained. The binary classification algorithm can be extended to solve multi-classification issues as well and one solution is OvO(One versus One). Assume there are data which have n categories. The idea of OvO is that pair up n categories to generate n(n-1)/2 binary classifiers. The n(n-1)/2 classifiers have n(n-1)/2 prediction results then the final classification result is generated by voting when new data come. In this lab the training data has four distinct labels which means there are four classes. The training data are used to train six binary classifiers and the final prediction is generated by voting when inputting test data. Apart from the OvO technique, OvM(One versus Many), MvM(Many versus Many) and softmax function are able to cope with multi-classification problems as well.

Data preprocessing is indispensable before building the classifier. The data standardization is used to preprocess the training data of logistic regression. The standardization formula is presented in figure3.4.

$$X' = \frac{X - mean}{\sigma}$$

figure3.4

X is the value of the sample data. mean is the average value of a single feature and σ is the standard deviation of a single feature. The effect of this method results in rescaling all data into a small range, which is beneficial for reducing the impact of the data value, data size, data distribution difference and data features on the models. The mathematical principle of logistic regression is illustrated above, the vital process of training logistic regression classifier is to calculate the value of θ and θ decides the upper limit of the classification performance of logistic regression. The values of training data directly affect calculating the value of θ, which indicates that the feature values count for logistic regression. Standardization is able to weaken the affection of large values of some certain features, which ensures every data feature can be learned by the model fairly. The optimization process of the cost function is accelerated after data standardization if gradient descent is used and it is potential to raise the accuracy rate.

The first step of the experiment is to build the classifier by Python API and intuitively grab the performance. The accuracy rate of the test set is 51.26 percent. The second step is to debug the parameters to attempt to optimize the performance of logistic regression. There are multiple parameters provided by the API. Two vital factors which could affect the performance of this classifier are multi-classification method and optimization algorithm of cost function then the two relevant parameters will be debugged. The result of cross validation showed that OvM(One versus Many) classification method and l-bfgs algorithm to optimize the cost function have the highest accuracy rate. Train the classifier with the optimal parameters and the test set is used to evaluate the classifier. The prediction results of the model with the input of the training set and the test set are obtained. The accuracy rates are both slightly higher than 50 percent which suggests the logistic regression has the problem of data underfitting. The cross-validation result of the accuracy is 51.57 percent which is still not high. The confusion matrix of logistic regression is shown in the figure3.5 and this figure clearly shows the classifier fails to correctly classify data that have label 0, 2 and 3. The value of recall and precision for data labeled 2 and 3 is 0. The precision value of data labeled 0 is 0.17 and the precision value of data labeled 1 is 0.49 according to the confusion matrix, which shows the performance is poor.

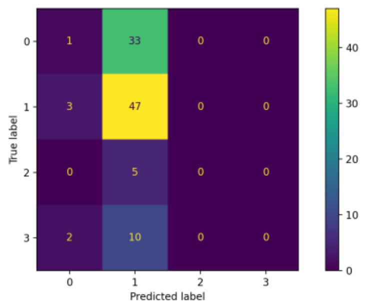

figure3.5

Run the code multiple times and the results found that the precision value of data labeled 1 is extremely unstable and none of the attempts the classifier managed to classify data into classes 2 and 3. Compute the value of auc and it is almost equivalent to 0.5 which shows the performance of logistic regression is similar to a random classifier. The classifier classifies almost all the data into class 1 because the majority of data in the original dataset is labeled 0 and 1, which indicates the logistic regression is affected seriously by the highly skewed dataset. The weight is given to the data to solve the problem of data imbalance. The more data in a certain data label, the lower weight. In this method, the accuracy of logistic regression dramatically decreases to 20 to 30 percent which is still not a well-performed classifier. Logistic regression is not a suitable model according to the evaluation above.

3.3 KNN

KNN is the abbreviation of k-nearest neighbor. The general idea of knn to classify data is comparatively simple. In general, the dataset is divided into a training set and a test set. Compare the new data to the data in the training set to find the similarity when the new data come. Take the k data that are most similar. Select the majority labels of the k data and that label is the label of the new data [2].

Consider the situation as follows. A dataset that has multiple features and one of the features has a range from 1000 to 10000, but the remaining features have a range from 0 to 1. The values of one certain feature are far larger than those of the rest features which causes the classification results to be dominated by one feature. The data standardization is used to preprocess the training data of knn and the principle of standardization has been elaborated in section 3.2. The value of data can be rescaled into the same range level after the standardization. The circumstance that one feature dominates the overall classification result will not occur after data preprocessing.

Build the classifier by the Python API and the accuracy of the test set is 43.02 percent on average. It is not a high accuracy rate then debug the significant parameters to improve the performance. The classification process of knn is based on calculating the similarity between data points, then the value of k and the measurement index of data similarity are two most important parameters that need to be debugged. The value range of k is generally taken from 2 to 20 to debug [2]. The result of cross validation showed that k=20 and chebyshev similarity calculation method have the highest accuracy rate. Train the classifier with the optimal parameters selected in the debugging process. The test set is split to evaluate the model performance. The accuracy rate of predicting the training set is close to 60 percent and the accuracy rate of predicting the test set is slightly higher than 50 percent. It concludes that knn exists the same problem as logistic regression which is data underfitting. The accuracy rate of test set prediction used cross validation is 50.2 percent which is promoted. The confusion matrix of knn is shown in the figure3.6 and the matrix has shown clearly that knn classify all data into class 0 and 1. The recall value and precision value of data labeled 2 and 3 is zero. The precision value of data labeled 0 is 0.47 and the precision value of data labeled 1 is 0.52 according to the confusion matrix. The performance is relatively poor but slightly superior to logistic regression.

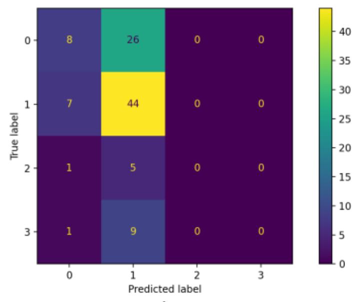

figure3.6

Run the code multiple times the knn model still cannot correctly predict data labeled 2 and 3. The data points in the provided dataset have a promiscuous distribution and no clear pattern existed, which means the distribution of the data points labeled differently is not clustered. The vast majority of data is labeled 0 and 1 in the dataset. The data labeled 0 and 1 have a large probability of overlapping with data that have distinct labels. The principle of knn is to compute the similarity between data points to realize classification. It concludes that the highly imbalanced dataset and the chaotic data distribution influence the performance of knn in a high probability on the basis of its algorithm principle and the experiment results, which indicates knn may not fit for this specific task. Debug the parameters to attempt to solve the issue of the original dataset. Assign a weight to each data and the data weight is inverse to the distance. The cross validation showed that the accuracy rate is 50 percent which is nearly the same as the previous ones, but the accuracy rate of the training set dramatically raise up to above 90 percent which may lead to the problem of data overfitting. The attempt to solve the imbalanced issue failed. The auc value is 0.53. The overall conclusion is that knn is not a suitable classifier as well but knn is slightly superior to logistic regression.

3.4 Decision Tree

Literally, decision tree completes the classification task by constructing a tree structure. The decision tree is able to be generated by distinct algorithms that contain ID3/4/5, CART and CLS [4]. ID3 is taken as an example to illustrate the classification principle of the decision tree. Entropy and information gain are two vital factors of the ID3 algorithm used to build the overall decision tree [4]. The entropy of the target needs to be computed at first. It splits the dataset on certain attributes of features into tables. In the case of this lab, the original dataset contains features Q1, Q2, Q3, Q4 and Q5. The label is students' major. The dataset is divided into subtables, which are major and Q1, major and Q2, major and Q3, major and Q4, major and Q5. The entropy of every subtable needs to be calculated to obtain the corresponding information gain. The attribute which has the highest information gain is selected to be the decision node. Recursively split the dataset to construct a complete decision tree. Simply put, the general idea is to find the most efficient decision order in terms of the training dataset to classify data. The constructed decision tree will classify data when new data come.

The classification process of decision tree is different from logistic regression and knn as elaborated above. The model constructing of Logistic regression and knn relies on the value of training set data points. The construction process of decision tree structure does not depend on the numerical value of data. The data is not further preprocessed as the numerical value is not the decisive factor of the model performance.

Build the classifier by the Python API and the default value of the branching criterion is Gini Impurity which in general has similar performance to entropy, then Gini index has been used in this lab. The test set is divided to evaluate the model performance. The accuracy rate of predicting the training set is 96.5 percent. The accuracy rate of predicting the test set is 40.84 percent by using cross validation. The accuracy of the training set is close to 100 percent but the test set accuracy remains at a low level, which suggests the decision tree has a problem with data overfitting. The parameter controlling the maximum depth of the tree needs to be debugged to alleviate data overfitting. Set the integer value from 3 to 15. The test set accuracy cannot reach a high level however the selection of max depth. The highest accuracy rate appears where the maximum depth is 3. Train the classifier with the training set. The maximum depth is set to 3. The auc value is 0.53. The accuracy rate of the training set decreases to 55 percent and the accuracy rate of the test set increases to 51.78 percent using cross validation, which causes data underfitting. The confusion matrix is shown in figure3.7. The recall value and precision value of data labeled 2 and 3 is zero. The precision value of data labeled 0 is 0.35 and the precision value of data labeled 1 is 0.54 according to the confusion matrix.

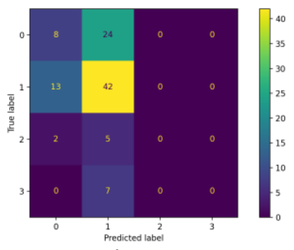

figure3.7

The precision value and recall value of data labeled 2 and 3 indicate that the decision tree has a common problem with previous classifiers. The decision tree fails to correctly classify data labeled 2 and 3. Readjust the maximum depth of the tree attempting to find a balance between performance and classification. The performance is relatively high and data labeled 2 and 3 can be slightly classified when the maximum depth is 6. The confusion matrix is shown in figure3.8.

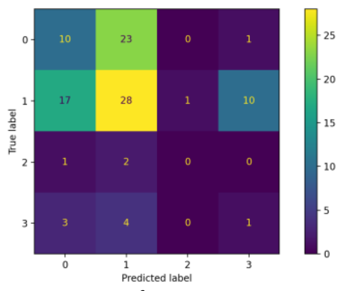

figure3.8

F1-score of data labeled 0 is 0.31 and f1-score of data labeled 1 is 0.5. The f1-score of data labeled 2 is 0 but the f1-score of data labeled 3 is 0.1. The cross validation is used to attain the accuracy rate of the test set which is 47.76 percent and the accuracy rate of the training set is 66.83 percent. The accuracy rate decreased as compared to the previous one. However, the advantage of the decision tree is able to classify data labeled 2 or 3. The auc value is computed to be 0.56 which is higher than the previous one but the value is still close to 0.5. There is no significant improvement in accuracy rate, recall, precision and f1-score of the decision tree as compared to logistic regression and knn. The auc value among the three classifiers is close to 0.5 and various evaluation indexes are low which means the logistic regression, knn and decision tree perform poorly. The unique ability of the decision tree is to slightly allevaite the classification issues caused by the highly imbalanced dataset if parameters are debugged appropriately. The last attempt is to build the random forest classifier which is an upgraded and more complicated version of the decision tree.

3.5 Random Forest

Random forest can be intuitively considered as a collection of multiple decision trees but each tree is constructed slightly diversely. Random forest ultimately chooses the majority decisive result of all trees as the final result for a classification task [5]. In the case of this lab, the original dataset is divided into the training set and the test set. Assume there are N samples and M features in the training set. Select one sample among the N samples randomly and repeat N times to form a new training set which is used to construct the tree. Select m features randomly at every node and restrict that m << M. The value of m is invariable in the process of building the random forest [5]. Following the above steps, grow every single tree to form the forest.

Random forest is an ensemble method and is a collection of several decision trees. The data is preprocessed the same way as the decision tree classifier which is illustrated in section 3.4. No further processing needs on the input data.

Build the classifier by Python API. The accuracy rate of the test set using cross validation is 44.6 percent. The accuracy of predicting the training set is 96.1 percent, which indicates that random forest has a problem with data overfitting. The maximum depth of the trees is limited to alleviate data overfitting. The number of decision trees in the forest affects the ultimate performance of the classifier. The parameter controlling tree number and the maximum depth is debugged to improve the classification effects. The maximum depth value is set as integers in the range from 3 to 10. The number of trees is set as a series of discrete values of 50, 100, 150, 200, 250 and 300. The accuracy rate reaches the highest when the maximum depth is 4 and the number of trees is 250. Train the classifier with the optimal parameters. The result shows that the classifier has the same problem as the previous classifiers, which is data underfitting. Readjust the maximum depth to cope with data underfitting. Setting the maximum depth to 6 to some degree mitigates data overfitting. The training set accuracy is close to 70 percent and the test set accuracy is 50.18 percent applying cross validation. The corresponding confusion matrix of the classifier is shown in figure3.9.

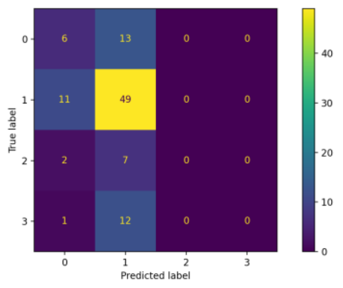

figure3.9

The confusion matrix shows that random forest with the parameters of maximum depth = 6 and tree number = 250 fails to classify data labeled 2 and 3. The f1-score of data labeled 2 and 3 is 0. The f1-socre of data labeled 0 is 0.31 and the f1-score of data labeled 1 is 0.7 which is the highest among the attempted classifiers. Run the code multiple times. Random forest classifier has predicted correctly data that is labeled 2 or 3 in rare times. The auc value of the classifier is 0.574 which is slightly higher than the decision tree. In conclusion, random forest has a common problem with logistic regression and knn, which classify the majority of data into class 2. Random forest has superior performance to logistic regression and knn based on the evaluation indexes, precision, recall, f1-score and auc value. The f1-score of data labeled 1 is 0.7 which indicates that the classification of data labeled 1 performs well. However, it performs poorly on the classification of data labeled not 1.

3.6 recommend one classifier

Four different classifiers have been constructed in this lab. The original dataset is used as input among the four distinct classifiers. The same input is used for all classifiers, which helps to contrast their performance. The auc value of the four classifiers is close to 0.5 which indicates poor performances. The accuracy rates of the test set are lower than 60 percent. Logistic regression and knn have the data underfitting problem. Decision tree and random forest have the problem of data overfitting. The evaluation indexes used in the experiment are precision, recall and f1-score. The evaluation index values of the four classifiers are scarcely higher than 0.6. It concludes that none of the classifiers have excellent performance. In this lab, a special dataset is provided. The data distribution is not clustered and it can scarcely find efficient data bias, which is difficult for the classifiers to have excellent performance. The experiment results have proved that logistic regression classifies almost classifies all data into class 1. The highly imbalanced dataset has completely affected the performance of logistic regression. The accuracy rates of logistic regression and knn are similar but knn is able to classify data into classes 0 and 1. The classification task conducted by knn is based on the similarity between data. Theoretically, the imbalanced dataset contains most data labeled 1 and the data lack significant bias. It may cause the similarity between new data and data labeled 1 to be the highest in most cases. The experiment results proved knn performs similar to logistic regression. The first two classifiers, logistic regression and knn are both not recommended. Decision tree can slightly mitigate the undesirable influence of the original dataset. The accuracy rate, precision, recall and auc are close and even higher than the first two classifiers. Random forest has the same classification problem as the first two that classifies almost all data into classes 0 and 1. The difference is that random forest classifies data labeled 0 and 1 more accurately. Decision can classify data labeled 2 and 3 but in extremely low performance. The accuracy rate, f1-score and auc value of random forest are higher as compared to the decision tree. The ultimate recommendation classifier is random forest after overall consideration and evaluation.

4. Unsupervised Classifications

One significant difference from supervised learning is that data have no labels in unsupervised learning. Organizing data in some specific structures is the goal of unsupervised learning [6]. The task is to cluster data with an unsupervised algorithm. Intuitively, Clustering means grouping data into clusters. Agglomerative clustering which is a bottom-up unsupervised method is attempted to cluster in this lab.

4.1 feature selection

The visualization result of TSNE is shown in figure1.5. The data distribution shown in the visualization result of TSNE is well-clustered by observation if data labels are ignored, which is helpful for the clustering task. In addition, the clustering results can be visualized intuitively if the input dataset is reduced to two dimension. TSNE is chosen to be the input of the agglomerative clustering algorithm with comprehensive consideration.

4.2 Agglomerative Clustering

The general idea of agglomerative clustering is bottom-up. Assume that n data points are used as input. Setting the ultimate clustering goal is to group input data into k clusters or set the final distance threshold. Initially, there are n clusters. Each of the points m1, m2, m3, …, mn is regarded to be a single cluster Kj = {mj}. The algorithm computes the similarity between the clusters. Multiple mathematical methods can be used to calculate cluster similarities, such as Euclidean distance and cosine similarity. Iterate the following step until it has existed k clusters or the distance threshold is reached. Compute and find two most similar clusters K_a and K_b, then remove clusters K_a and K_b and the union $K_a \cup K_b$ is the new cluster which need to append into the existed set of clusters [6].

The detailed process of how the students are grouped into clusters is as follows. The TSNE is used to reduce the data dimensionality which ultimately obtains two features as the input of the algorithm. Each sample point after dimensionality reduction records the mark information of one unique student, which suggests each point can be regarded as one student. There are n samples as input and the goal is to obtain k clusters. Each point p1, p2, …, pn is assigned to an individual cluster. Euclidean distance is used to compute the cluster similarity. Generally, multiple linkage criteria can measure the cluster similarity. The complete linkage is applied among multiple criteria in this lab. Iterate the following procedures. Calculate the Euclidean distance between samples in sub-clusters and the maximum values are taken as the distance between two clusters. Select two clusters Ki and Kj which has the minimum distance. These two clusters are merged into one new cluster and Ki and Kj are removed. Terminate the iteration process until there are k clusters already. Set the integer k in the range of 2 to 6. Visualize the result of the clustering with different values of k. Compute the NMI value and silhouette score and visualize the two evaluation indexes in a line chart.

4.3 visualization results and evaluation

The clustering visualization results with distinct values of k are shown in figure4.1. The line chart of the evaluation indexes is shown in one graph in figure4.2.

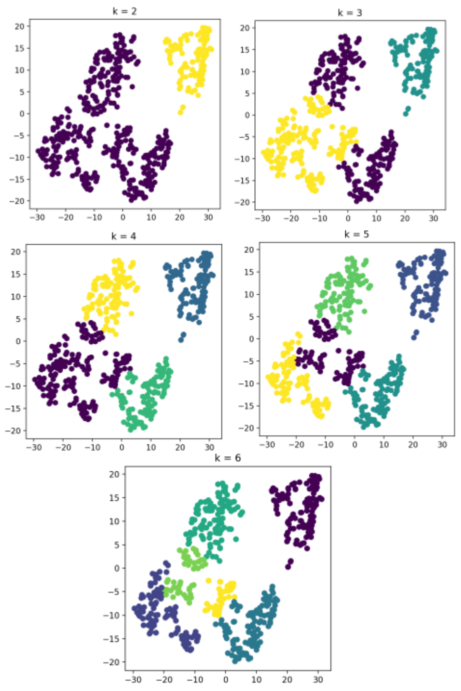

figure4.1

Grouping data into 4 clusters means that k = 4 intuitively has the optimal clustering effect based on the five visualization graphs in figure4.1.

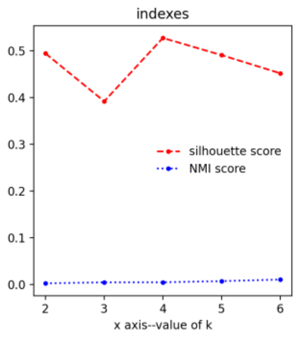

figure4.2

The silhouette score achieves the highest when k = 4 and the NMI score increases tiny as k increases. It can regard that NMI scores retain almost the same whatever the value of k. 4 clusters have the optimal effect according to both visualization results and evaluation indexes.

5. Conclusion

Three tasks have been completed in this lab. The first task is feature extraction. A highly skewed and imbalanced dataset is provided. Several basic features such as average value, medium value and standard deviation are extracted and visualized after data preprocessing. It ultimately found that the basic features lack significant data bias. Three complex mathematical operations which are PCA, TSNE and NMF have been attempted to find useful data bias but they failed. The final decision is to directly use the original dataset as the input of the classifiers. The second task is building classifiers in a supervised way. Four classifiers have been built, which are logistic regression, knn, decision tree and random forest. Logistic regression classifier and knn are affected seriously by the skewed and imbalanced and they have poor performance, which concludes that logistic regression and knn are not recommended. Decision tree and random forest have poor performance as well but they perform better than the first two classifiers. Most of the evaluation indexes of random forest are superior to the decision tree. Ultimately, random forest is the recommended classifier to finish the second task. The third task is building one classifier by an unsupervised algorithm. Agglomerative is selected to cluster data. The dataset obtained by TSNE is the input of the algorithm. Both the visualization results and the evaluation index values indicate that the optimal effect reaches when grouping into 4 clusters.

References:

[1] C. Shah, "A Hands-On Introduction to Data Science". Cambridge University Press, n.d., pp.235-236

[2] C. Shah, "A Hands-On Introduction to Data Science". Cambridge University Press, n.d., pp.248

[3] C. Shah, "A Hands-On Introduction to Data Science". Cambridge University Press, n.d., pp.237

[4] C. Shah, "A Hands-On Introduction to Data Science". Cambridge University Press, n.d., pp.253

[5] C. Shah, "A Hands-On Introduction to Data Science". Cambridge University Press, n.d., pp.261

[6] C. Shah, "A Hands-On Introduction to Data Science". Cambridge University Press, n.d., pp.291