INT104 Coursework 2 Report

Zhibo He 2141913

Lab-D1/4-Group-A

May 16, 2023

1 Introduction

1.1 Requirements

In order to provide more comfortable medical care, it is necessary to consider the development of more rational methods of anesthesia for different patients. By collecting 5,000 patients with a 15-question physical condition questionnaire, it is possible to categorize the patient's physical condition and develop a variety of anesthesia methods appropriate to the particular case.

1.2 Process and Results

In this research paper, the authors will demonstrate the research procedures from pre-processing data operations, such as data cleaning and dimensionality reduction, to the construction of supervised and unsupervised learning classifiers.

The PCA dimensionality reduction algorithm was used to reduce the number of features from fifteen to **9 features** with the highest feature importance.

The authors construct three supervised classifiers: support vector machines, random forests, and Multi-layer perceptrons. **The Multi-layer Perceptron** Classifier was then chosen as the final supervised learning classifier, judging the degree of fit based on the magnitude of the evaluation metrics of the classifier's prediction results, such as accuracy, roc curve, F1 score, and other values.

The authors built a **K-means** clustering model for unsupervised learning. The classification clusters were selected as 2 clusters by SSE validation and silhouette score, so the patient's health status data were finally clustered into **two clusters**.

2 Dimensionality reduction

2.1 Data Cleaning

The author must select the correct data to build classifier models according to the requirement. In order to achieve that, it is necessary to have a data pre-processing operation. First and foremost, the author uses *pd. Read* method to get the row dataset from the given CSV document. Then, since the result only has values '0' and '1', the author deletes the data labeled as '2'. After that, the author uses *drop(column[])* method, putting the dataset into two pieces, 'data' and 'label.' The result of data cleaning is printed in Figure 1. As you can see, after deleting the '2' data, the data value decreases from '5344' to '5330', and the question results, also written as data, also decrease from '17' columns to '15' columns. The author considers each column of the data as the data feature dimension.

```
Task1:
value amount of Dataframe , columns:  (5344, 17)
value amount of Dataframe , columns:  (5330, 17)
value amount of Data , columns:  (5330, 15)
```

Figure 1:

2.2 Principal Component Analysis

As required, a reasonable method needs to be found that minimizes the loss of information contained in the original indicators while reducing the number of indicators to be analyzed in order to achieve a comprehensive analysis of the data collected. As there is a certain correlation between the variables, it is possible to combine the various types of information present in each variable separately using a smaller number of composite indicators. *Principal Component Analysis (PCA)* is one such method of dimensionality reduction. In the process of PCA, high-dimensional data is downscaled by linear transformation while retaining the maximum data variance. In addition, the original features are projected onto the main feature directions in the data to achieve data dimensionality reduction and de-redundancy. Furthermore, in order to be able to perform Principal Component Analysis, it is necessary to choose reasonable components for the analysis (Shlens, 2014).

Based on these requirements, the author drew heat maps of the data features to determine the degree of relevance of the data features. The color band colors are mapped to the heat map matrix data in the heat map. In general, the closer the color is to the positive value, the higher the expression and positive correlation, while the closer the color is to the negative value, the lower the expression and negative correlation.

Figure 2 shows *the heat map* of the data feature dimensions. The amount of data is **too large and bloated**, resulting in very poor visualization. Therefore, the author decided to use a random sampling approach by selecting '0.005' of the dataset as the down-sampling rate, and based on the selected sample data, the visualization was analyzed. The value of 'random state' was chosen to be '42' to make it possible to repeat the experimental data.

According to the new version of *the heat map* in Figure 3, it can be seen that the degree of correlation of the data decreases from left to right on the horizontal axis. However, the heat map also has the obvious disadvantage that the magnitude of the features is

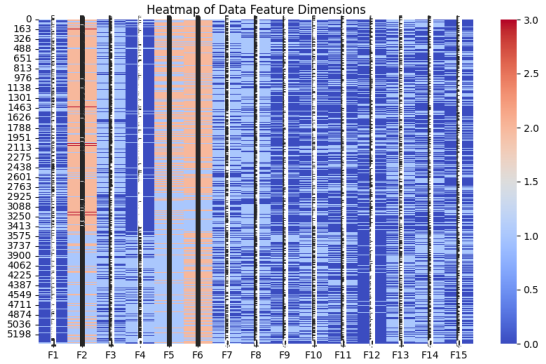

Figure 2:

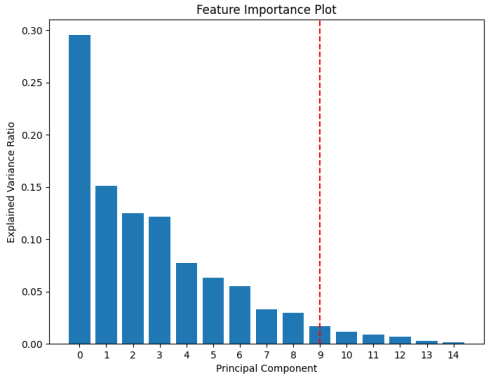

Figure 4:

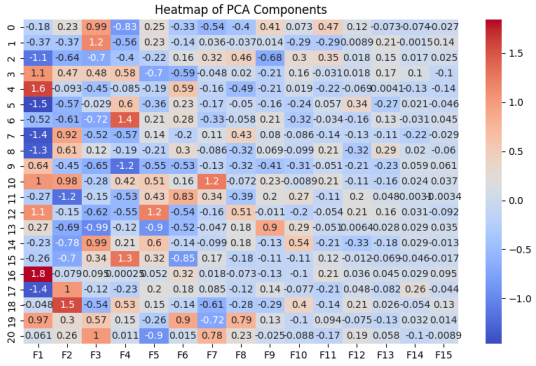

Figure 3:

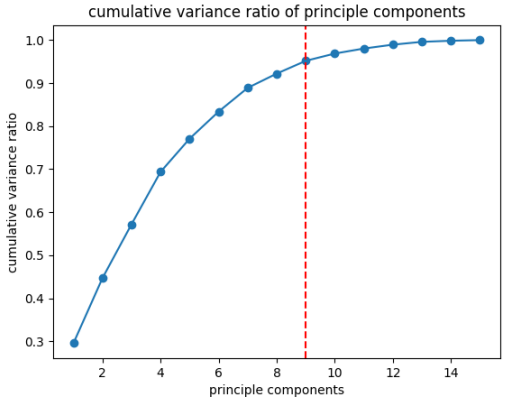

Figure 5:

expressed in different colors and can be chosen more subjectively, so a more objective way of selecting values for the components is needed.

With the *'explained variance ratio'* variable, the interpretable variance ratio of each feature can be obtained, and by calculating it, the author obtained the image of Figure 4. As can be seen from the figure, the first feature contains 0.296 of the variance of the dataset, which is a relatively large percentage, and the additional features contain less dataset variance.

In order to be able to balance data completeness and complexity, the author finds the elbow point that explains 0.95 of the variance portion of the data by calculating the sum of the data variances. In Figure 4 and 5, respectively, the red dashed lines show that, according to the image of Figure 5, when the dimensionality of the data is 9, the degree of loss for the explainable variance is smaller, and the dimensionality reduction effect is better. Therefore, the author chose to reduce the original data to **9 dimensions.**

3 Training Classifiers in a Supervised Way

According to the requirements, the author needed to build at least three supervised learning classifiers when the data pre-processing was completed. In order to complete the classification requirements and judge the accuracy of the machine learning classification algorithm, and also to avoid over-fitting, the author chose **the K-fold** method of *cross-validation*, dividing the data and labels into a training set and a test set, 'test size' parameter is set as '0.2', and the value of k set to 5, so that four samples were used for training and one sample for validation at a time. The data were used in the training set for classification, and the results were verified in the test set. Next, the data that had been downscaled to 9 dimensions using PCA was applied to the training set data using the *'fit transform'* method.

3.1 Support Vector Machine Classifier

First and foremost, *Support Vector Machine* is used to classify samples by mapping the data to find a hyperplane with maximum interval(Chang and Lin, 2011). The hyperplane acts as a decision boundary that maximizes the distance between different classes of data points known as support vectors. The SVM is suitable for solving problems in high-dimensional complex situations.

For constructing the support vector machine classifier model, the author first created an SVC object *(SVM classifier)* from sklearn.svm package. Because of the relatively small size of the training set, the author chose RBF (radial basis function) as the kernel function. After several attempts, the author found that the training results of the SVC model were poorly fitted, so the author set the gamma parameter to 0.01 and the C hyperparameter to 15. By decreasing the gamma parameter and increasing C, the author widened the bell curve and increased the influence range of each instance to make the decision boundary flatter.

Next, the author used the fit function to train the training data (data after PCA dimensionality reduction), where the transform function was used to transform 'X test' to 'X test' pca and 'X train' to 'X train pca' according to the cross-validation results. Is the test data and training data after PCA reduction to 9 dimensions, and 'Y train' is the 'X train' pca corresponding to the training labels.

Finally, the author uses the trained SVM model to predict the reduced-dimensional test data and the predict function to generate the prediction results stored in 'Y pred'.

3.2 Random Forest Classifier

As a supervised learning method, *the Random Forest* integrates decision trees, in which each decision tree is constructed by bootstrap sampling and feature randomization of the training data. The random forest performs the classification by voting or averaging the predictions of each decision tree. In addition, in a random forest, the random forest searches for the best features in a randomly generated subset of features, providing superior robustness and randomness (Ho, 1998).

Random forests can also measure the relative importance of each feature. This index measures how much each feature contributes to the predictive power of the classifier and can be obtained using Scikit-Learn's *'feature importances'* variable. For this dataset, based on the scores printed in Figure 7, it can be seen that the decision forest has been learned to obtain higher importance of around '0.30' for features 1 and 4 and less than '0.05' for the rest.

For the training process of this classifier model, the author first created a RandomForestClassifier and set some parameters. The 'n estimators' (the number of decision trees in the random forest) was set to 200, and the 'max depth' (the maximum depth of each decision tree) was set to 5. The model was validated by taking the values several times to ensure that the model was as accurate as possible without overfitting. Then, the random forest classifier was trained using the training set' Y train', label, using the dimensionality reduction data' X train pca' from PCA. Lastly, the trained model is used to predict the test data, 'X test pca', to obtain the corresponding prediction, 'Y pred'.

3.3 Multi-layer Perceptron classifier

A neural network classifier is a supervised learning algorithm using an *Artificial Neural Network (ANN)*. Where the input layer receives feature data, the hidden layer uses elements such as weights to transfer and learn features, and the output layer generates the classified results (McCulloch and Pitts, 1990). Neural networks are capable of solving non-linear and complex classification problems.

Multi-Layer Perceptron (MLP), one of the simplest ANN architectures, can be used for binary classification tasks. The results that exceed a threshold are computationally classified into positive classes and vice versa into negative classes.

The author created a multi-layer perceptron classifier *(MLPClassifier)* where the 'hidden layer sizes' parameter was set to (10, 5) and 'max iter' was set to 200, which was found by Iterative optimization used to find the local best-fit result. Next, the 'X train pca' and 'Y train' are trained using the *fit* method of the multi-layer perceptron classifier. The trained model is then used to predict the test set data' X test pca' to obtain the 'Y pred' prediction.

3.4 Evaluation for Classifiers

For the three supervised learning classifier models mentioned in the previous section, the author first evaluates the fit of the machine learning classification results by finding *the ROC curve*, which is suitable for the evalu-

ation of dichotomous classification tasks, and visualizes the presentation of the results by predicting the correctness or otherwise of the classification of the data into positive and negative values compared to the original data.

However, when the author used the ROC curve to analyze the SVC model as an example, as shown in Figure 6, the curve was too smooth, and the *AUC (Area Under Curve)* was relatively small, making it a poor fit and difficult to use as a reasonable evaluation criterion.

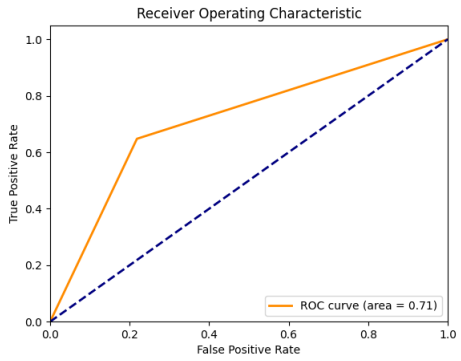

Figure 6:

As shown in Figure 7, the performance of the three supervised learning classification models did not significantly diverge, with the MLP classifier being slightly better. Therefore, the author chose **MLP** model as the supervised learning classifier.

```
Task2:
For SVC model:
SVC Accuracy: 0.6885553470919324
SVC Recall: 0.6142857142857143
SVC Precision: 0.6779279279279279
SVC F1 Score: 0.6445396145610279
SVC Specificity: 0.7517361111111112
SVC AUC: 0.6830109126984129


For Random Forest Classifier model:
[0.34725655 0.03149896 0.04340016 0.30217618 0.04573737 0.03770993
 0.05874986 0.03586913 0.09760188]
RFC Accuracy: 0.6744840525328331
RFC Recall: 0.5877551020408164
RFC Precision: 0.665127020785219 3
RFC F1 Score: 0.6240520043336946
RFC Specificity: 0.7482638888888888
RFC AUC: 0.6680094954648526


For neural network classifier model:
MLP Accuracy: 0.7310037523452158
MLP Recall: 0.6326530612244898
MLP Precision: 0.6652360515021459
MLP F1 Score: 0.6485355648535566
MLP Specificity: 0.7291666666666666
MLP AUC: 0.6809098639455784
```

Figure 7:

4 Unsupervised Classification

For the selection of the unsupervised learning classifier, the author chose the *K-means* model, which is a simple and fast clustering model that only needs to determine the number of clusters K to make it try to learn the clustering centers of different clusters without adding other parameters. It is, therefore, extremely critical to find a suitable value of k when using the K-means algorithm for clustering analysis.

For building the K-means clustering model, the author first introduced the original data and used the *'range()'* method to delineate the values of k as positive integers from 1 to 10, which were stored in the increasing array K. The K-means clustering model was then built. Next, a K-means clustering model is built with any k of K as the parameter choice, 'n init' is the default value of 10, and 'random state' is chosen as 42 to increase the repeatability and testability of the model. The fit operation is then used to allow the model to learn X train pca, and the 'inertia' variable is applied to calculate the SSE (Sum of Squared Errors) value of the K-means clustering model, which is repeated iteratively until the k value is 10.

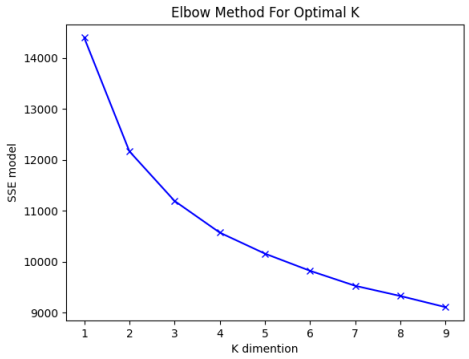

Figure 8:

Figure 8 shows the *SSE (Sum of Squared Errors)* values for the K-means algorithm for k in the interval (1, 10). The SSE gives a clear visual indication of the inertia of the model (the mean square distance of each instance from the nearest centroid) for different values of k and, thus, the degree of dispersion of the clustering model. The SSE will generally have a steeply slowing inflection point, known as the elbow, where the corresponding k-value is chosen for the best clustering results. Thus, in Figure 8, clustering works best when $\mathbf{k = 2}$ is chosen. However, it is undeniable that using SSE to determine the number of clusters is rather crude, and a more precise approach needs to be found.

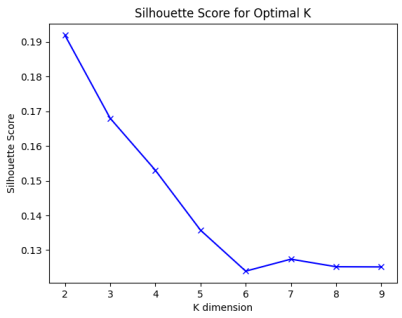

Figure 9:

The author set the number of clusters more rigorously by using the *silhouette score*. The silhouette score is the average silhouette coefficient of all instances. The closer the coefficient is to one, the better the clustering, with each instance being closer to a particular cluster center and further away from the other cluster centers. A negative value indicates a high probability that the instances are entering the wrong cluster to be assigned to. As can be seen from Figure 9, the silhouette coefficients fluctuate down from small to large values of k, again indicating that classification is best at $\mathbf{k = 2}$.

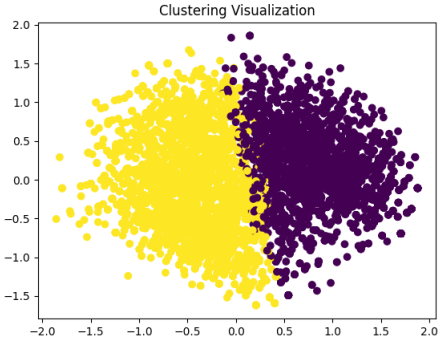

Figure 10:

Therefore, based on the best number of clusters chosen, the author rebuilt a k-means clustering model with k=2 and used the appropriate method to make the model learn the 'X train pca' data and predict the output predictions based on the learning results 'y kmeans', and store the labels of k-means. Finally, a two-dimensional scatter plot was created to represent the clustering results. The results show that two clusters with distinct boundaries are formed by k-means clustering.

5 Conclusion

5.1 Results

In this lab report, according to the requirements, the author completed the construction of the clustering model from data pre-processing to the classifier, using the PCA algorithm down to **9 dimensions**, deciding to use the **Multi-layer perceptrons** classifier based on multiple assessment criteria and finally using the k-means clustering model to classify the patient's physical condition into **two clusters**, showing that two different methods of anesthesia can be developed to improve medical care.

5.2 Perspectives and inspiration

For the data pre-processing aspect, it is a good idea to use PCA to reduce the dimensionality in the face of complex data with many dimensions. Heatmaps can be drawn for dimensional selection to find the more relevant dimensions by color comparison. However, although intuitive, heatmaps are not a rigorous way of thinking about dimension selection. Therefore, images such as explained variance ratio cumulative variance can be drawn to visually compare the importance of features in different dimensions, which is more appropriate when the explained variance ratio is 0.95. A dimensionality around this avoids overfitting and reduces the dimensionality of the data to a large extent.

In addition, random forests can also show how much each dimension contributes to the prediction accuracy, so random forest models can be nested into the PCA data dimensionality reduction process to increase the robustness of the dimensionality reduction algorithm.

The evaluation criteria for ROC curves are not fully applicable in every case. When the values of both predicted and actual data are 0 and 1, the probability calculation of the truth of the predicted data may be wrong, resulting in a smoother and over-fitted ROC curve.

Cross-validation is always an excellent way to prevent overfitting. When the amount of data is small, the K-fold algorithm can divide the training and test sets and then build the machine-learning model.

6 Reference

Shlens, J. (2014)*A tutorial on principal component analysis, arXiv.org.* Available at: https://arxiv.org/abs/1404.1100 (Accessed: 15 May 2023).

Chang, C., Lin, C. (2011). *LIBSVM.* ACM Transactions on Intelligent Systems and Technology, 2(3), 1–27. https://doi.org/10.1145/1961189.196119 (Accessed: 15 May 2023).

Ho, T. K. (1998). *The random subspace method for constructing decision forests.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8), 832–844. https://doi.org/10.1109/34.709601

McCulloch, W. S., Pitts, W. (1990). *A logical calculus of the ideas immanent in nervous activity.* The Bulletin of Mathematical Biophysics, 5(4), 115–133. https://doi.org/10.1007/bf02478259

Arthur, D., Vassilvitskii, S. (2007). *k-means++: the advantages of careful seeding.* In Symposium on Discrete Algorithms (pp. 1027–1035). https://doi.org/10.5555/1283383.1283494