Principal component analysis machine learning to determine the membership of globular cluster M56

Jiahui.Li
UCL
MSc Scientific and Data Intensive Computing

Declaration

I, Jiahui.Li, confirm that the work presented in this dissertation is my own. Where information has been derived from other sources, I confirm that this has been indicated in the dissertation

Repository

jimmylihui/project: using PCA to analyse star clustering (github.com)

Abstract

Rishel et al have presented a preliminary study of stellar membership of the M56 globular cluster based on positions and proper motions of stars. As they admitted, the conclusion is not rigorously treated as the population of stars is small. Besides, he did not give the range of proper motions and positions for a member. In this article, I revisit the data and apply modern principal component analysis and machine learning algorithms to build a more precise model to predict the membership of the global cluster and apply the model to a larger size of data from Gaia collaboration. During the building of the model, parameters of principal component analysis and machine learning algorithms are compared and refined. The range of principal component to determine a member is illustrated by the boundary. The effect of principal component analysis on the accuracy and efficiency of the model is verified. Finally, the result of members in M56 from Gaia collaboration is given by the model.

Contents

List of Figures **3**

List of Tables **3**

1 introduction **4**

1.1 background about globular cluster Messier 56 4

1.2 Motivations 4

1.3 Objectives 5

1.4 Structure 5

2 Literature Review **5**

2.1 Principal component analysis 5

2.2 Globular cluster 6

2.3 Field star 6

2.4 Standard scalar 6

2.5 Artificial neural network 7

2.6 Support vector machine 7

3 Methodology **8**

3.1 Data loading and processing 8

3.2 Criterial establishment 9

3.3 Data Visualization 9

3.4 Applying PCA to the sample 9

3.5 Applying SVM to model and refining parameters 9

3.6 Testing model in other zones and retraining model 10

3.7 Training model using ANN and verifying the effect of PCA 10

3.8 Applying model to new data 10

4 Result **11**

4.1 Visualization of data 11

4.2 Visualization of principal components 11

4.3 Visualization of SVM model and parameters refinement 12

4.4 Testing SVM model and model retraining 14

4.5 Training model using ANN and verifying the effect of PCA 15

4.6 Application of model in latest data of M56 16
5 Limitation 18
6 Conclusion 18
7 Appendix 18
8 Reference 21

List of Figures

Figure 1: structure of the neural network……7
Figure 2: Illustration of SVM……...8
Figure 3: Visualization of data with spatial information……11
Figure 4: Two-dimensional principal components using different kernels……12
Figure 5: SVM classification using different sets of parameters……..12
Figure 6: The validation accuracy of different gamma and C……..13
Figure 7: visualization of the boundary in the inner 4 zones with principal components……14
Figure 8: The process and training set score and test set score of the ANN model……20
Figure 9: loss function of ANN models with PCA and without PCA……...15
Figure 10: The procedure of keras……16
Figure 11: The principal components of data from Rishel et al [1]……..17
Figure 12: The principal components of data from Gaia Collaboration [2]……17

List of Tables

Table 1: The f1 score, precision score, and recall score of different parameters……13
Table 2: The f1 score, precision score, and recall score of other zones……..14
Table 3: The f1 score, precision score and recall score in the testing set……14
Table 4: f1 score, precision score, and recall score of four zones……15
Table.5: Running times for MLP with PCA and without PCA……16

1 introduction

1.1 background about globular cluster Messier 56

The globular cluster Messier 56 (M56) was firstly discovered by Charles Messier on January 19, 1779. It is in the constellation Lyra and at about 32,900 light-years from Earth. Its combined mass 230,000 times that of the Sun. It is around 31-32 kilolight-year from the Galactic Centre and 4.8 kilolight-year above the Galactic Centre. The age of M56 is estimated to be 13.70 billion years. The stars with the highest brightness in M56 are of 13 magnitudes. Because the last stages of stellar evolution for low-mass stars are most likely observed in globular clusters, it is essential to distinguish field stars projected onto the cluster and cluster members randomly. For example, Rishel et al (1981) [1] have compiled a list of 39 potential cluster stars in the field of M56, convolving relative proper motions with spatial information. Harris et al (1982) extend the work of Rishel et al (1981) and conclude that most radial-velocity cluster members are confined to the giant branch, and the most UV-bright stars with detected velocities are field stars.

1.2 Motivations

Although Rishel et al (1981) bring up a new criterion of convolving relative proper motions with spatial information to determine the member in M56, he admitted the conclusion is not rigorously treated as the population of stars is small. Besides, he did not give the specific range of proper motions and positions to determine a membership. Hence it would be helpful to build a model using modern algorithms and high-performance computers to extract the range of proper motions and positions from his prediction, and apply the model to the larger amount of data to verify the correctness of the criterion.

In recent years, the emergence of machine learning and principal component analysis (PCA) has improved the efficiency and accuracy of solving classification problems. PCA is the most widely used feature extraction method. Machine learning is the field of study that trains a computer to learn without being explicitly programmed. PCA and machine learning are applicable to identify members of M56 because of that PCA can be used to reduce the dimensionality of data, such as proper motions and spatial information to improve the efficiency of classification, and machine learning can solve the problem of classifying members with given training data. Typical algorithms of machine learning include support vector machines (SVM) and artificial neural networks (ANN). The former classifies objects by drawing boundaries and the latter classifies objects by using the weighted sum and the activation function.

In this report, four types of data of stars in M56 are used to identify member stars of M56, which are x, y position (mm); individual proper motion in x and y. The data is collected from Rishel et al (1981) [1] and member stars for training are also collected from it. Then PCA is applied to reduce the dimensionality of data to two. SVM and ANN are applied to train the model. During the procedure, the

effect of parameters in PCA and machine learning algorithms is tested. Finally, the model is applied to the latest data of stars in M56 from Gaia Collaboration (2018) [2]. The classification result is viewed both in plot and in text.

1.3 Objectives

The purpose of this project is to build a model using principal component analysis and machine learning algorithm and apply it to the larger amount of data in M56 to obtain members of globular cluster. During the project a few sub objectives need to be met:

- The data should be collected from Rishel et al [1] and Gaia collaboration [2]
- Parameters of the model should be adjusted to provide the best performance of the model
- The boundary of the model should be illustrated
- The accuracy of the model should be verified
- The effect of PCA on the model should be discovered
- The model should apply to the data from Gaia collaboration and generate a list of members

1.4 Structure

Following the introduction, Chapter 2 will introduce the background knowledge of terms being used in this project. Chapter 3 will display the process of the project, it shows the process of collecting and loading data, the adjustment of parameters for the model, illustrating the boundary of the model, verifying effect of principal component analysis, and applying the model to new data. Chapter 4 illustrates the result of process in chapter 3 with explanation to the result. Chapter 5 explains the limitation of the project, and the cause of the limitation. Chapter 6 evaluates the project with brief conclusion.

2 Literature Review

2.1 Principal component analysis

Principal component analysis (PCA) is a mathematical algorithm that aims to reduce the data dimension while reserving most of the variation in the data set. Reduction is achieved by identifying directions, called principal components, where the variation in the data is maximal along directions. Each sample can be expressed by a few principal components instead of many variables. In the mathematical form, for a $n \times p$ data matrix X with zero mean by columns, where each of the n rows represents an individual object and each of the p columns represents a type of feature. The one-dimensional principal component scores $t_{(i)} = (t_1, \dots t_l)_{(i)}$ is given by:

$$t_{k(i)} = X_{(i)} \cdot W_{(k)} \ \ for \ \ i = 1, \ldots, n \ \ \ k = 1, \ldots, l \quad [1]$$

Where $W_{(k)} = (w_1, \ldots, w_p)_k$ is p dimensional vectors of weights.

The first component is calculated by:

$$W_{(1)} = arg\ max\{\frac{W^T X^T X W}{W^T W}\} \quad [2]$$

The k_{th} component is calculated by:

$$\hat{X}_k = X - \sum_{s=1}^{k-1} X W_{(s)} W_{(s)}^T \quad [3]$$

$$W_{(k)} = arg\ max\{\frac{W^T \hat{X}_k^T \hat{X}_k W}{W^T W}\} \quad [4]$$

The maximum values for the quantity in brackets are given by their corresponding eigenvalues

2.2 Globular cluster

Globular clusters are tight groups of million stars grasped together by their mutual gravitational attraction, with a nearly spherical distribution and high density in the centre.

In a globular cluster, individual star motions are determined by the sum of the mass of all stars within the cluster.

2.3 Field star

A randomly located star that lies along the line of vision to a group of physically associated stars under study, such as a star cluster. Field stars are not connected with an astronomical object being learned. These field stars are necessary to identify to prevent contamination in the study

2.4 Standard scalar

Standard scalar is a tool to help with standardising a dataset which involves rescaling the distribution of values so that the mean of observed values is 0 and the standard deviation is 1. A value is standardized as follows:

$$y = (x - mean)/standard\ deviation$$

Where mean is calculated as:

$$mean = sum(x)/count(x)$$

standard deviation is calculated as:

$$standard\ deviation = \sqrt{\sum \frac{(x-mean)^2}{count(x)-1}}$$

2.5 Artificial neural network

Artificial neural networks (ANNs) [3] are computational networks inspired by biology. Among the various types of ANNs, we focus on multilayer perceptron (MLPs) with back propagation learning algorithms [4]. MLPs which are mostly used for a wide variety of problems, are based on a supervised procedure and comprise three layers: input, hidden, and output. Its simplest form can be observed in Figure 1.

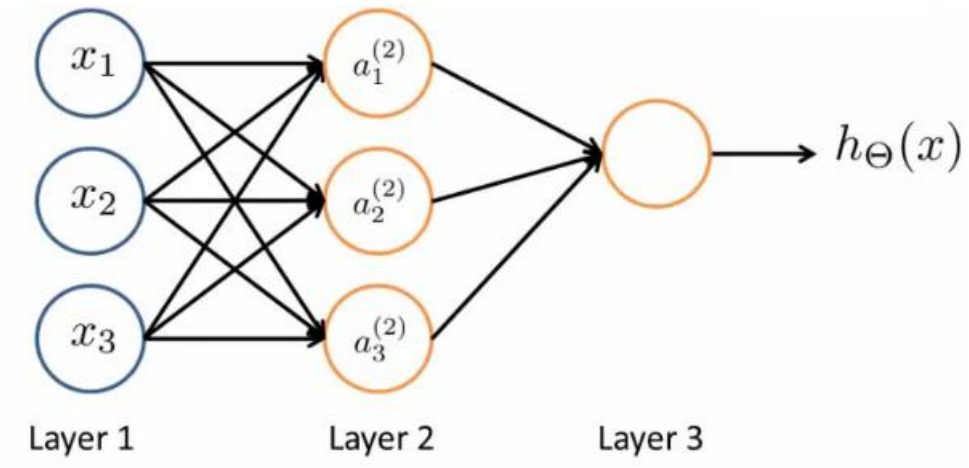

Figure 1: structure of the neural network

The output in layer 2 is computed by the weighted sum of the previous layer, see Equation 5, and the activation function, see Equation 6.

$$Z_j = \sum_{i=1}^{n} \theta_{ij} x_i \quad [5]$$

Where Z_j is the weighted sum and i is the number of the input and j is the number of the output.

$$a_j = h(Z_j) \quad [6]$$

Where a_j is the j_{th} output and h is the activation function.

2.6 Support vector machine

Support Vector Machine (SVM) is a supervised learning algorithm that analyzes data for classification and regression analysis [7]. It classifies the objects by constructing a hyperplane that has the largest distance to the nearest training data point of any category (functional margin). In other words, it aims to maximize the margin. Its illustration can be seen in Figure 2.

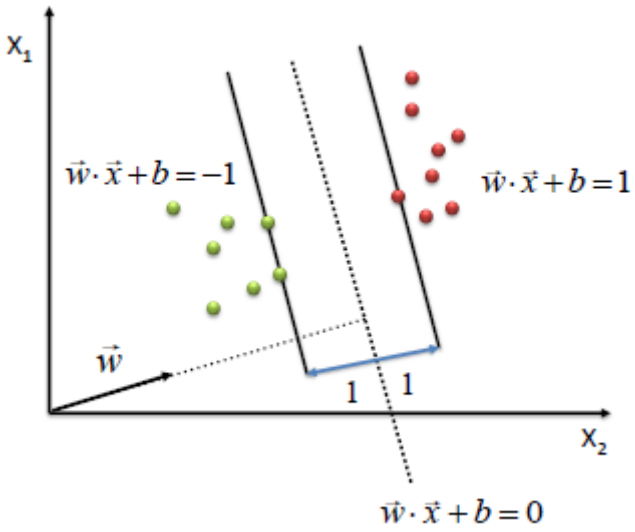

Figure 2: Illustration of SVM

In Figure 2, there are two hyperplanes represented by Equation 7 and Equation 8.

$$W^T X + b = 1 \quad [7]$$

$$\mathrm{W^T X} + b = -1 \quad [8]$$

Where W is the normal vector to the hyperplane. X is the p dimensional vector.

The distance between two hyperplanes is $\frac{2}{|w|}$. Hence maximizing margin equals minimizing |w|. The optimization of the problem can be represented by Equation 9.

$$minimizing\ |w|\ subjects\ to\ y_i(W^T X + b) \geq 1\ for\ i = 1, \ldots, n \quad [9]$$

Where y_i is the class for each point.

It can be observed that W and b are only decided by X which lies nearest to the hyperplane. These X are called support vectors.

3 Methodology

The work is processed on python 3.8 with the assistance of the colab notebook.

3.1 Data loading and processing

Since the original data are typed from Rishel et al, it is necessary to load them into digital form for processing. Data loading is achieved using Excel. The full data can be seen in Appendix 1. It is necessary to notice that Rishel et al only label the stars in the inner four zones. To identify those stars. I firstly find the centre of data by minimizing the total distance from one star to other stars. Then stars are classified into zones due to their distance to the centre. Standard scalar is applied to the data for improving the applicability of the model.

3.2 Criterial establishment

In the original data, there are six parameters included for each star. The x,y position(mm); S(microns), the std. deviation of the Gauss function describing the density of the stellar image, measuring stellar brightness; Proper motion in x and y in arcsecond per century. And the total weight.
To identify the membership, only the x, y position and proper motion in x and y are needed. The other 2 parameters are discarded. The central four zones are 0-5mm,5-7.07 mm, 7.07-8.66mm, and 8.66-10.00 mm from the centre separately.

3.3 Data Visualization

Hence the data is four-dimensional, it is hard to represent it in a figure intuitively. I decide to plot the data with their spatial information: The x,y positions. The class of data points is separated by their colors. The blue color indicates it is not a cluster member and the red color indicates it is a cluster member. The plot is done by pyplot, a collection of functions that draw the plot on python.

3.4 Applying PCA to the sample

PCA needs to be initialized before using it. Several parameters need to be set: the number of principal components, the type of kernel, gamma and C if necessary. I initialize three PCAs with different kernels [5] with corresponding parameters [6] to discover the effect of kernels. The first one uses a linear kernel. The second one uses an rbf kernel with gamma=0.04. The third one uses a sigmoid kernel with gamma=0.001, coef0=1. To visualize the difference more clearly, the scatter plot of the principal component is presented for each PCA. The PCA of best performance is chosen for the following project.

3.5 Applying SVM to model and refining parameters

SVM [7] also needs to be initialized before using. The kernel for SVM is the radial base function kernel. The critical parameters for it are gamma and
C [8]. Four pairs of gamma and C are set: (0.1,0.001), (0.1,1000), (10,0.001), (10,1000). Each pair of parameters is used to initialize an SVM which is applied to two-dimensional principal components in Zone 1 extracted from PCA. The performance of SVM is described by three scores: f1 score, precision score, and recall score [9]. The value of those scores is proportional to the performance of SVM. Plots of classified results using each parameter are also drawn to present the effect of parameters intuitively.
Functions StratifiedShuffleSplit [10] and GridSearchCV() [11] can be used exhaustive search over

specified parameter values for an estimator. The function is used to search the C in the log space of (-2,10) and gamma in the log space of (-9,3) for an SVM applied in Zone 1. The performance is valued by validation accuracy, which describes how accurate the model is in the testing set.

3.6 Testing model in other zones and retraining model

The performance of the model training in Zone 1 can be observed by testing it in other zones. The process of testing is to enter principal components in other zones into the model and obtain the predicted class which is compared with the actual class. The performance is evaluated by the f1 score, precision score, and recall score. Plots of the classification are also drawn by pyplot.

Then the model is retrained by the data in the inner four zones to improve the performance. Forty percent of the data is used as a training set and the other sixty percent is used as a testing set. Firstly, the model is fitted with a training set, then it is tested in a testing set to give the evaluation of the scores. The boundary of the model is also illustrated.

3.7 Training model using ANN and verifying the effect of PCA

ANN is applied to train the model. Python provides a package called MLPclassifier to build a multilayer perceptron classifier. Parameters of (hidden_layer_sizes=(50,), max_iter=100, alpha=1e-4,solver='sgd', verbose=10, tol=1e-4, random_state=1,learning_rate_init=.1) are used to initialize the MLPclassifier. The training set is forty percent of the data in the inner four zones, and the testing set is the other sixty percent of that. The performance is evaluated on the f1 score, precision score, and recall score. The code refers to J.McEwen et al [12].

The effect of PCA is verified by comparing the performance of ANN with PCA and that without PCA. The effect is verified from three perspectives: accuracy, efficiency, and loss function [13]. Accuracy is evaluated with training scores and testing scores. Efficiency is evaluated with time spent on processing the training. The loss function measures the absolute difference between the prediction and actual value, and it is presented with a plot. The time is calculated by a function called timeit. The parameters of the MLP classifier and the training and testing sets remain.

An alternative option for ANN called Keras [14] is shown. Keras is connected by a two-dimensional input layer, a fully connected dense layer with 300 neurons, including a ReLU activation function, another fully connected dense layer with 100 neurons, including a ReLU and a final dense layer with 10 neurons. The model is compiled by setting the loss function to sparse categorical cross-entropy with a stochastic gradient descent optimiser. The code refers to J.McEwen et al [15].

3.8 Applying model to new data

The latest data of stars in M56 is collected from Gaia Collaboration [2], covering a range of 4 arcminutes around the centre of the cluster. The new data is processed with a standard scalar at first, to reduce the error from scales of arcminutes. Then two-dimensional principal components are extracted from the data using PCA. Then those principal components are entered into a model trained by the data from Rishel et

al. The model gives a predicted class for each principal component.

4 Result

4.1 Visualization of data

The data of M56 from Rishel et al are visualized according to their spatial information and class, without considering the proper motion. Each data point in Figure 3 represents a star, and the red color indicates it is a cluster membership while the blue color indicates it is not a cluster membership [1]. The centre of M56 is at (20.1255, 20.002499999999998). The visualization can be observed in Figure 3.

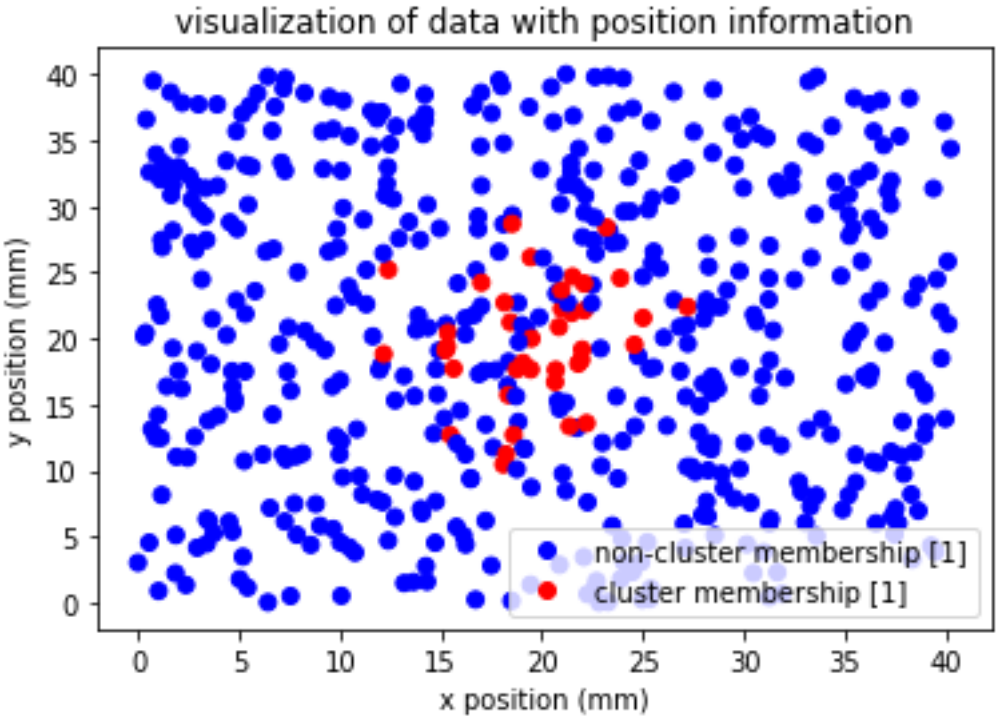

Figure 3: Visualization of data with spatial information

it can be observed membership stars are gathered in the central zone.

4.2 Visualization of principal components

The principal components for PCAs with multiple kernels and corresponding parameters can be observed in Figure 4.

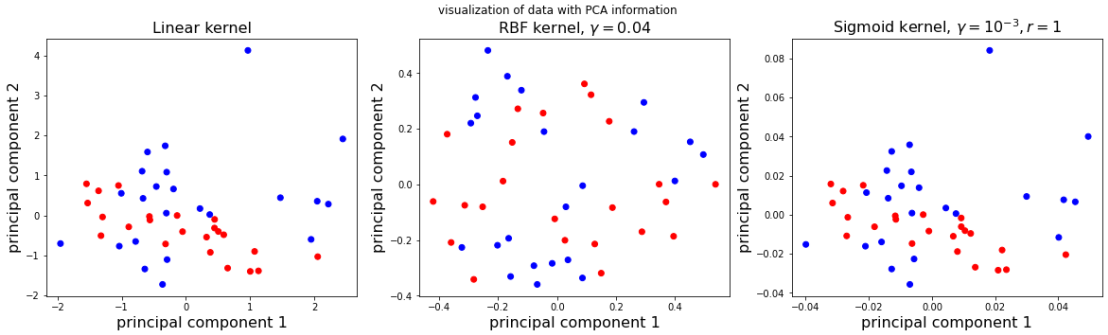

Figure 4: Two-dimensional principal components using different kernels

In the case of the sigmoid kernel, data points of member stars are mostly separated from the non-member stars. This is convenient for the separation. Hence, in the following part of the report, PCA with the sigmoid kernel, and gamma=0.001, coef0=1 is used.

4.3 Visualization of SVM model and parameters refinement

The boundary of Models trained with SVM in Zone 1 with four pair of parameters: (0.1,0.001), (0.1,1000), (10,0.001), (10,1000) is shown in Figure 5.

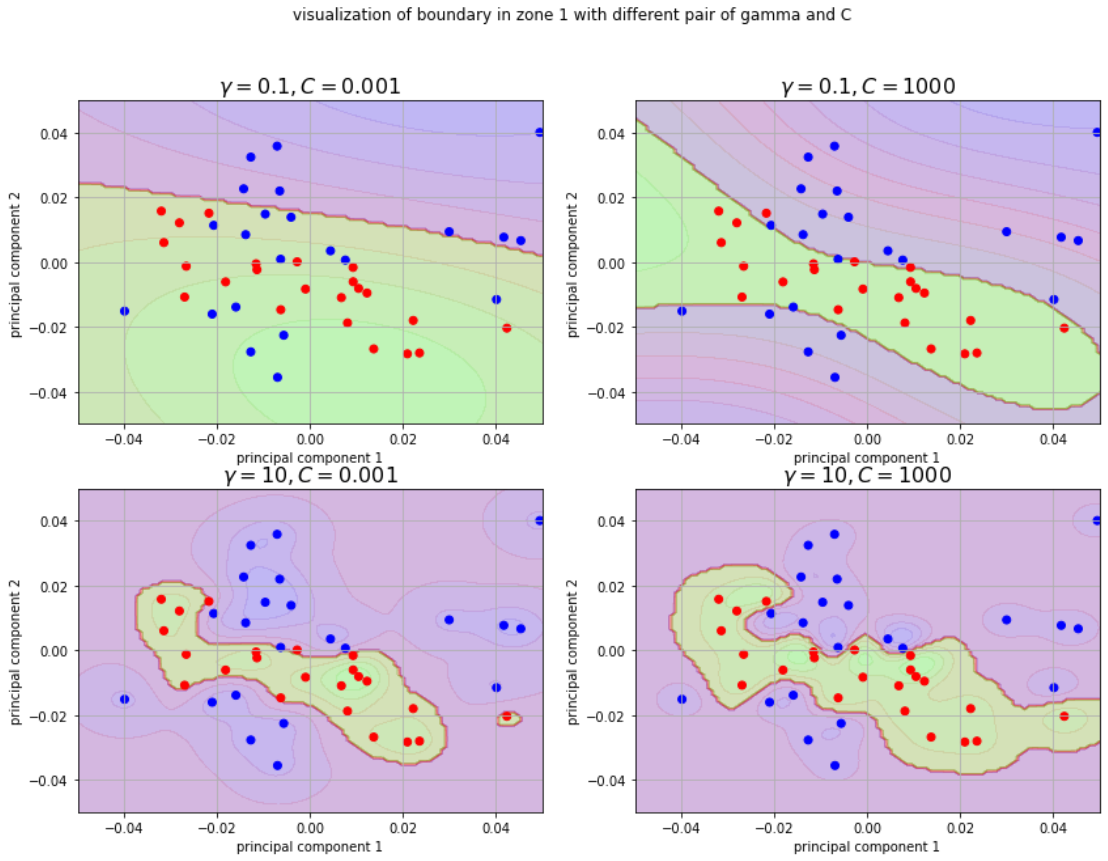

Figure 5: SVM classification using different sets of parameters

The f1 score, precision score, and recall score of models in different parameters can be seen in Table 1.

Table 1: The f1 score, precision score, and recall score of different parameters

parameters	F1 score	Precision score	Recall score
(0.1,0.001)	0.31	0.23	0.5
(0.1,1000)	0.94	0.94	0.94
(10,0.001)	0.31	0.23	0.5
(10,1000)	0.63	0.69	0.65

Hence the parameter (0.1,1000) gives the best performance. For the following part of the report, the parameter of SVM is set to (0.1,1000).

The plot of validation accuracy versus gamma and c can be seen in Figure 6. The orange block indicates the highest validation accuracy, and the red block is not presented in Figure 6 because accuracy with all parameters is above 0.5.

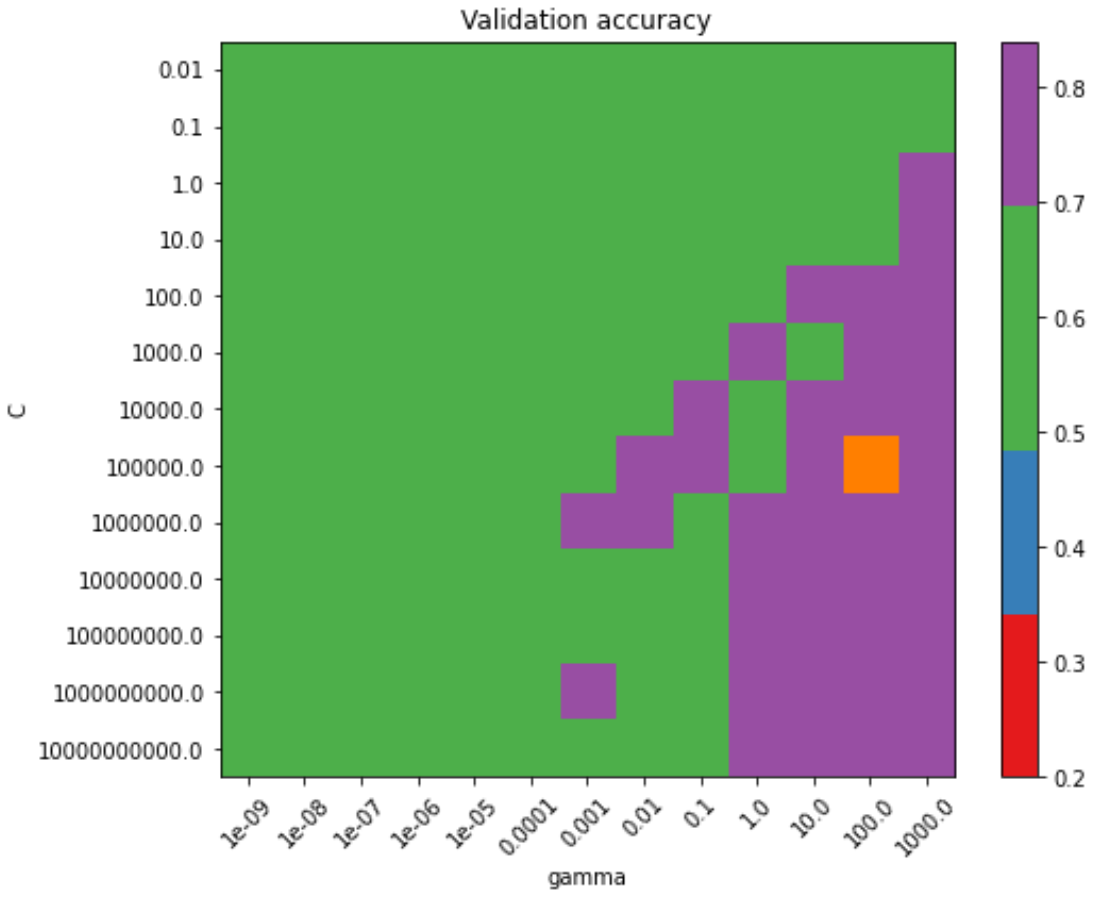

Figure 6: The validation accuracy of different gamma and C

The optimum parameter is {'C': 1000000.0, 'gamma': 100.0}. This pair parameter is not chosen for SVM because it is comparable slow for the training.

4.4 Testing SVM model and model retraining

The f1 score, precision score, and recall score of the model in other zones can be observed in Table 2. The model performs best in Zone 2 and worst in Zone 3.

Table 2: The f1 score, precision score, and recall score of other zones

parameters	F1 score	Precision score	Recall score
Zone 2	0.70	0.73	0.81
Zone 3	0.53	0.57	0.67
Zone 4	0.65	0.64	0.77

The SVM model is retrained with the principal components in the inner four Zones. The f1 score, precision score, and recall score in the testing set can be observed in Table 3. The scores are around 0.7.

Table 3: The f1 score, precision score and recall score in the testing set

parameters	F1 score	Precision score	Recall score
Testing set	0.70	0.70	0.73

The boundary of the classifier can be observed in Figure 7. The boundary gives the indication of criterion from Rishel et al to determine members in M56 convolving the proper motions and positions.

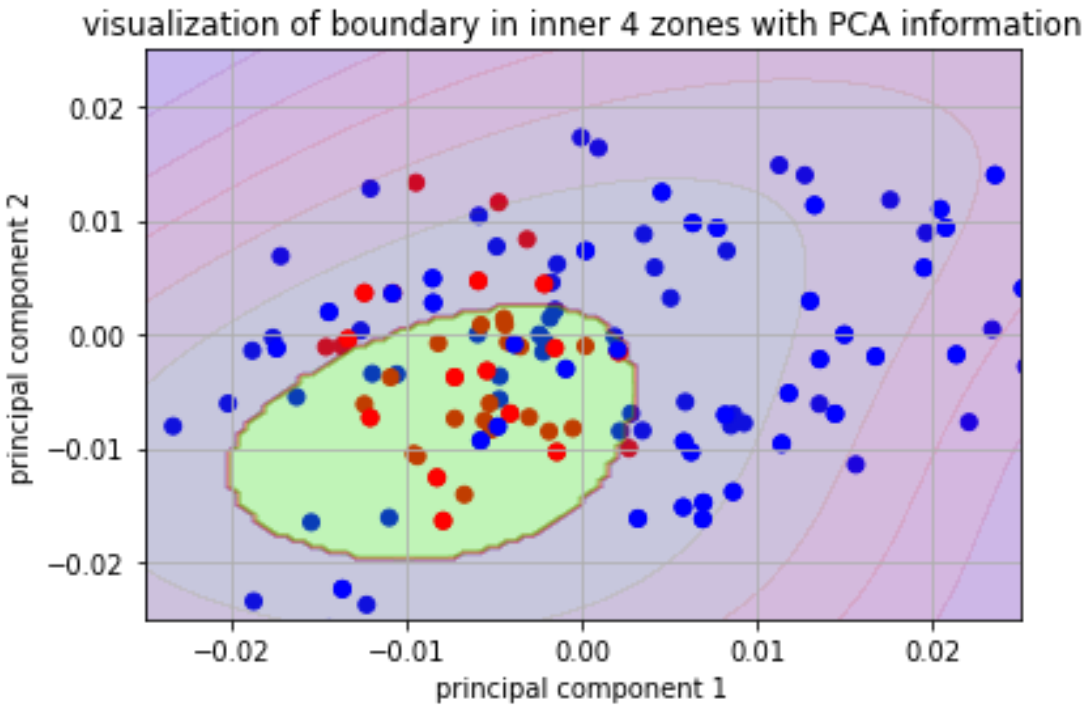

Figure 7: visualization of the boundary in the inner 4 zones with principal components

The model is applied to four Zones separately. The f1 score, precision score, and recall score in each zone can be observed in Table 4. The scores have been improved by increasing the size of the training

set.

Table 4: f1 score, precision score, and recall score of four zones

parameters	F1 score	Precision score	Recall score
Zone 1	0.69	0.70	0.69
Zone 2	0.74	0.72	0.76
Zone 3	0.65	0.64	0.76
Zone 4	0.68	0.70	0.66

4.5 Training model using ANN and verifying the effect of PCA

The process and training set score and test set score can be observed in Figure 8, seeing appendix.
The training scores of ANN and SVM are both 0.7 which means the accuracy of both training models agrees.
The loss function of ANN models with PCA and without PCA can be observed in Figure 9. In the case of using PCA, the loss function starts at a lower value and decreases slower than that without using PCA.

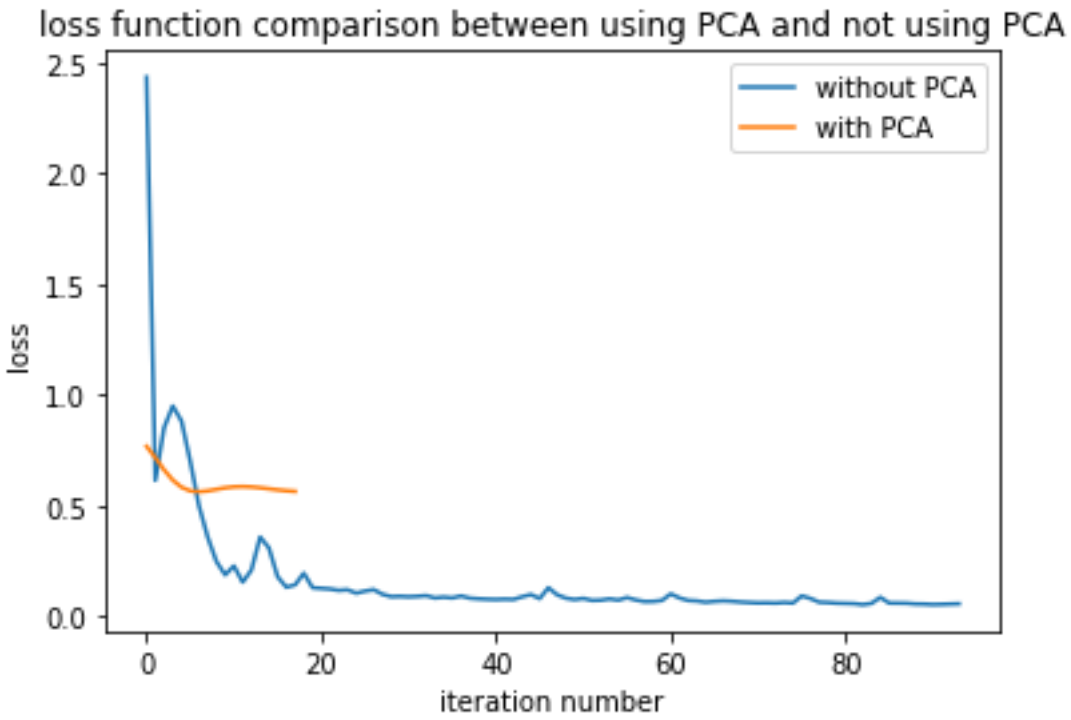

Figure 9: loss function of ANN models with PCA and without PCA

The training score of ANN without PCA is 0.94 and the testing score of ANN without PCA is 0.91. The training score of ANN with PCA is 0.77 and the testing score of ANN without PCA is 0.70. By comparing the loss function and training score of ANN models with PCA and without PCA, it can be concluded that the ANN without PCA has better performance. Because in the procedure of PCA, part of the information is lost which reduces the accuracy of models during training.

The procedure of keras can be seen in Figure 10.

```
Epoch 1/10
3/3 [==============================] - 0s 84ms/step - loss: 0.6910 - accuracy: 0.6111 - val_loss: 0.6865 - val_accuracy: 0.7083
Epoch 2/10
3/3 [==============================] - 0s 20ms/step - loss: 0.6836 - accuracy: 0.7500 - val_loss: 0.6799 - val_accuracy: 0.7083
Epoch 3/10
3/3 [==============================] - 0s 25ms/step - loss: 0.6757 - accuracy: 0.7500 - val_loss: 0.6745 - val_accuracy: 0.7083
Epoch 4/10
3/3 [==============================] - 0s 24ms/step - loss: 0.6692 - accuracy: 0.7500 - val_loss: 0.6699 - val_accuracy: 0.7083
Epoch 5/10
3/3 [==============================] - 0s 17ms/step - loss: 0.6637 - accuracy: 0.7500 - val_loss: 0.6655 - val_accuracy: 0.7083
Epoch 6/10
3/3 [==============================] - 0s 15ms/step - loss: 0.6582 - accuracy: 0.7500 - val_loss: 0.6606 - val_accuracy: 0.7083
Epoch 7/10
3/3 [==============================] - 0s 16ms/step - loss: 0.6523 - accuracy: 0.7500 - val_loss: 0.6567 - val_accuracy: 0.7083
Epoch 8/10
3/3 [==============================] - 0s 24ms/step - loss: 0.6476 - accuracy: 0.7500 - val_loss: 0.6530 - val_accuracy: 0.7083
Epoch 9/10
3/3 [==============================] - 0s 14ms/step - loss: 0.6426 - accuracy: 0.7500 - val_loss: 0.6482 - val_accuracy: 0.7083
Epoch 10/10
3/3 [==============================] - 0s 20ms/step - loss: 0.6366 - accuracy: 0.7500 - val_loss: 0.6444 - val_accuracy: 0.7083
```

Figure 10: The procedure of keras

The accuracy is of Keras 0.7, the same as the result of the ANN model and SVM model.
Running times for MLP with PCA and without PCA are shown in Table.5.

Table.5: Running times for MLP with PCA and without PCA

	MLP with PCA	MLP without PCA
Running time	0.10742	0.1584

The result obtained from Table.5 shows that in two dimensions, it saves 30% of running time by applying PCA, which is a considerable large resource saving.
It can be concluded that PCA saves 30% of running time at the cost of 23% of accuracy. It hints that in the situation where a large amount of data needs to be processed, and the resource is limited, PCA is a useful way to save resources. While if the resources are sufficient, PCA can harm the accuracy to a certain extent.

4.6 Application of model in latest data of M56

The principal components of data from Rishel et al [1] with labels are shown in Figure 11. The principal components of data from Gaia Collection with labels are shown in Figure 12.

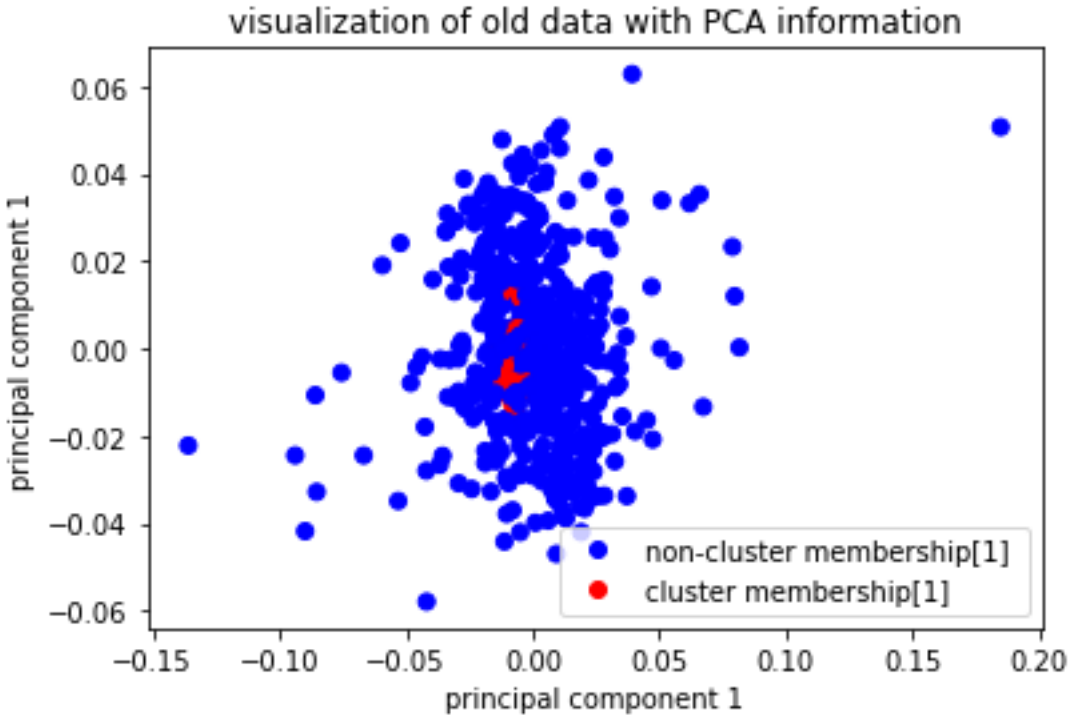

Figure 11: The principal components of data from Rishel et al [1]

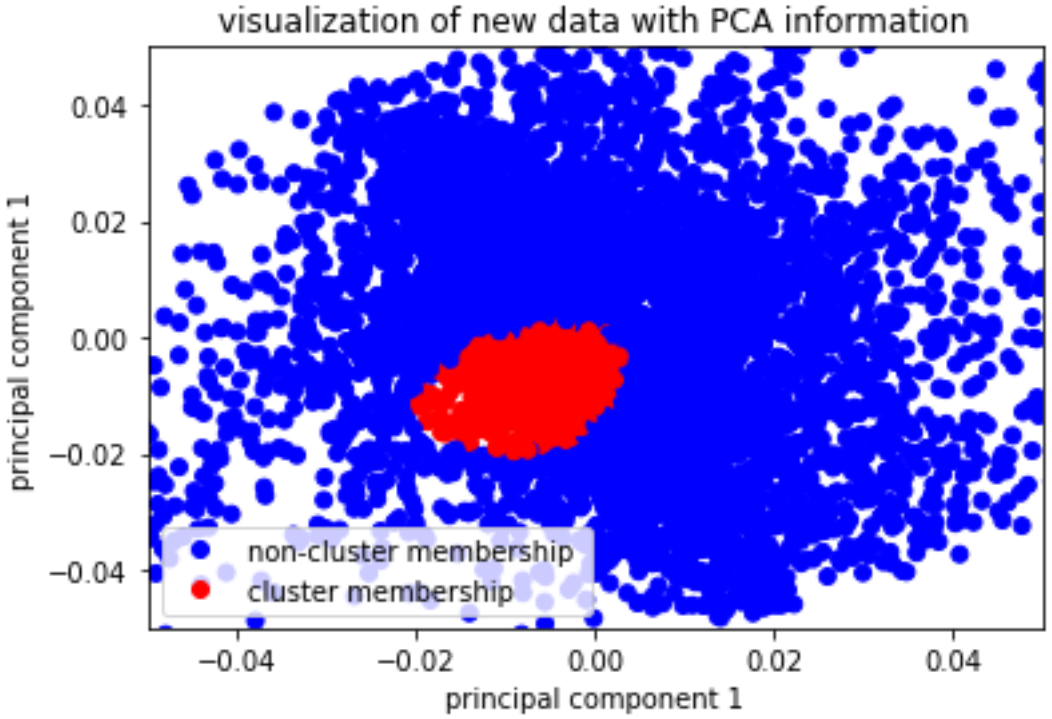

Figure 12: The principal components of data from Gaia Collaboration [2]

The decision boundary in the data of Gaia Collaboration is more distinctive than that in Rishel et al. This is because of that SVM tends to give a continuous boundary that separates the classes. There are 1117 stars classified as a member of the cluster out of the total number of 6677 in the data set of Gaia

collaboration. And in that of Rishel et al, the ratio is 39 out of 141. Ratios between cluster members and non-cluster members are close in two data sets.
the detail of the predicted class for data from Gaia Collaboration n is saved in a .txt file, see appendix.

5 Limitation

The study is built on the assumption that the prediction of Rishel et al is correct. Hence, the error in the prediction can propagate to the model. Besides, Gaia Collaboration does not provide the label of a star which determines if a star is a cluster member. It is hard to examine the accuracy of predicted labels with actual labels.

6 Conclusion

In this report, I apply the Principal Component Analysis to extract the principal components of stars in the M56 globular cluster from Rishel et al. Models are then obtained by applying the support vector machine algorithm and artificial neural network to the principal components. The boundary obtained from model gives the indication of criterion that Rishel et al used to determine the member of M56. The effect of PCA is also discovered. It saves 30% of running time at the cost of 23% of accuracy. Finally, the model using SVM is applied to the latest data of M56 from Gaia, and the prediction for each star is obtained. This study builds the models to identify the cluster star with statistical theory, instead of astronomical theory. It improves the speed of the classification for cluster members in M56 and provides an example for relative study in the future. While for the absence of actual labels in M56, it is hard to examinate to the accuracy of models.

7 Appendix

The repository is in jimmylihui/project: using PCA to analyse star clustering (github.com)
The code can be visited via project/Project.ipynb at main · jimmylihui/project (github.com)
The prediction result can be visited via project/prediction (1).txt at main · jimmylihui/project (github.com)
The old data of stars can be visited via project/data.txt at main · jimmylihui/project (github.com)
The new data of stars can be visited via project/project.xlsx at main · jimmylihui/project (github.com)
The process and training set score and test set score of ANN model is shown below:

```
Iteration 1, loss = 0.78737483
Iteration 2, loss = 0.71462950
Iteration 3, loss = 0.63926102
Iteration 4, loss = 0.57773329
Iteration 5, loss = 0.53649459
Iteration 6, loss = 0.51345862
Iteration 7, loss = 0.50369027
Iteration 8, loss = 0.50114044
Iteration 9, loss = 0.50071396
Iteration 10, loss = 0.49902095
Iteration 11, loss = 0.49457787
Iteration 12, loss = 0.48689036
```

Iteration 13, loss = 0.47630837
Iteration 14, loss = 0.46395359
Iteration 15, loss = 0.45147626
Iteration 16, loss = 0.43998364
Iteration 17, loss = 0.43003711
Iteration 18, loss = 0.42186854
Iteration 19, loss = 0.41525004
Iteration 20, loss = 0.40960656
Iteration 21, loss = 0.40485390
Iteration 22, loss = 0.40041454
Iteration 23, loss = 0.39619572
Iteration 24, loss = 0.39188157
Iteration 25, loss = 0.38769475
Iteration 26, loss = 0.38345967
Iteration 27, loss = 0.37935452
Iteration 28, loss = 0.37535805
Iteration 29, loss = 0.37148938
Iteration 30, loss = 0.36793397
Iteration 31, loss = 0.36472934
Iteration 32, loss = 0.36181626
Iteration 33, loss = 0.35914757
Iteration 34, loss = 0.35675461
Iteration 35, loss = 0.35449883
Iteration 36, loss = 0.35241063
Iteration 37, loss = 0.35050924
Iteration 38, loss = 0.34869246
Iteration 39, loss = 0.34696095
Iteration 40, loss = 0.34528314
Iteration 41, loss = 0.34365505
Iteration 42, loss = 0.34211623
Iteration 43, loss = 0.34064530
Iteration 44, loss = 0.33923201
Iteration 45, loss = 0.33783583
Iteration 46, loss = 0.33649448
Iteration 47, loss = 0.33521634
Iteration 48, loss = 0.33398655
Iteration 49, loss = 0.33282242
Iteration 50, loss = 0.33173672
Iteration 51, loss = 0.33065602
Iteration 52, loss = 0.32958844
Iteration 53, loss = 0.32856467
Iteration 54, loss = 0.32756794
Iteration 55, loss = 0.32655629
Iteration 56, loss = 0.32556942

Iteration 57, loss = 0.32461613
Iteration 58, loss = 0.32369815
Iteration 59, loss = 0.32275721
Iteration 60, loss = 0.32183393
Iteration 61, loss = 0.32091872
Iteration 62, loss = 0.32001603
Iteration 63, loss = 0.31913988
Iteration 64, loss = 0.31829419
Iteration 65, loss = 0.31746466
Iteration 66, loss = 0.31665960
Iteration 67, loss = 0.31600665
Iteration 68, loss = 0.31536527
Iteration 69, loss = 0.31472878
Iteration 70, loss = 0.31409671
Iteration 71, loss = 0.31346945
Iteration 72, loss = 0.31285541
Iteration 73, loss = 0.31224808
Iteration 74, loss = 0.31164181
Iteration 75, loss = 0.31105066
Iteration 76, loss = 0.31046816
Iteration 77, loss = 0.30992570
Iteration 78, loss = 0.30940892
Iteration 79, loss = 0.30889937
Iteration 80, loss = 0.30839504
Iteration 81, loss = 0.30789833
Iteration 82, loss = 0.30742066
Iteration 83, loss = 0.30695688
Iteration 84, loss = 0.30649691
Iteration 85, loss = 0.30602814
Iteration 86, loss = 0.30557616
Iteration 87, loss = 0.30514205
Iteration 88, loss = 0.30473120
Iteration 89, loss = 0.30432322
Iteration 90, loss = 0.30392562
Iteration 91, loss = 0.30352575
Iteration 92, loss = 0.30313822
Iteration 93, loss = 0.30280193
Iteration 94, loss = 0.30241525
Iteration 95, loss = 0.30206376
Iteration 96, loss = 0.30172296
Iteration 97, loss = 0.30137572
Iteration 98, loss = 0.30103058
Iteration 99, loss = 0.30071109
Iteration 100, loss = 0.30038086

Training set score: 0.824561
Test set score: 0.758621

Figure 8: The process and training set score and test set score of the ANN model

8 Reference

[1]Rishel, B. E., Sanders, W. L., and Schroder, R., “Membership in the field of globular cluster M 56.”, Astronomy and Astrophysics Supplement Series, vol. 45, pp. 443–450, 1981.

[2] Gaia Collaboration, 2018 Astron. Astrophys. 616, A1.

[3] S. A. Khaparde, P. B. Kale and S. H. Agarwal, "Application of artificial neural network in protective relaying of transmission lines," Proceedings of the First International Forum on Applications of Neural Networks to Power Systems, 1991, pp. 122-125, doi: 10.1109/ANN.1991.213509.

[4] K. Faez and M. Kamel, "Image reconstruction from contour data using a back-propogation neural network," Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing, 1994, pp. V/297-V/300 vol.5, doi: 10.1109/ICASSP.1994.389473.

[5] Wen Ge, Xu Hongzhe, Zheng Weibin, Zhong Weilu and Fu Baiyang, "Multi-kernel PCA based high-dimensional images feature reduction," 2011 International Conference on Electric Information and Control Engineering, 2011, pp. 5966-5969, doi: 10.1109/ICEICE.2011.5778352.

[6] Chen, Song Xi. "Probability density function estimation using gamma kernels." *Annals of the Institute of Statistical Mathematics* 52.3 (2000): 471-480.

[7] Z. Hao, L. Shaohong and S. Jinping, "Unit Model of Binary SVM with DS Output and its Application in Multi-class SVM," 2011 Fourth International Symposium on Computational Intelligence and Design, 2011, pp. 101-104, doi: 10.1109/ISCID.2011.34.

[8] H. Huang, Z. Wang and W. Chung, "Efficient parameter selection for SVM: The case of business intelligence categorization," 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), 2017, pp. 158-160, doi: 10.1109/ISI.2017.8004897.

[9] Ajitesh Kumar 2022, *Data Analytics*, accessed 26 july 2022, <https://vitalflux.com/accuracy-precision-recall-f1-score-python-example/>

[10] maryamnadeem20 2021, *Sklearn.StratifiedShuffleSplit() function in Python*, accessed 26 july 2022, <https://www.geeksforgeeks.org/sklearn-stratifiedshufflesplit-function-in-python/>

[11] Scott Okamura 2020, *GridSearchCV for Beginners*, accessed 26 july 2022, <https://towardsdatascience.com/gridsearchcv-for-beginners-db48a90114ee>

[12] Jason McEwen, Tom Kitching, Anastasis Georgoulas 2022, *Lecture 10: Artificial neural networks (ANNs)*, accessed 2022, <http://www.machinelearningwithbigdata.org/book/Lectures/Lecture10_ANN.html>

[13] Jason McEwen, Tom Kitching, Anastasis Georgoulas 2022, *Lecture 12: Introduction to Keras*, accessed 2022, <http://www.machinelearningwithbigdata.org/book/Lectures/Lecture12_IntroToKeras.html>

[14] D. Tejakumar, Mahardi, I. -H. Wang, K. -C. Lee and S. -L. Chang, "Predicting Surface Roughness using Keras DNN Model," 2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), 2020, pp. 338-341, doi: 10.1109/ECICE50847.2020.9301928.

[15] Seb 2021, *An introduction to Neural Network Loss Function*, accessed 2022, <https://programmathically.com/an-introduction-to-neural-network-loss-functions/>