Binary Classification Based on KNN Algorithm

Huanxin Sheng 15220202202189

January 17, 2023

Abstract

Machine learning is divided into supervised learning and unsupervised learning. For a manifold binary classification problem, both the classification algorithm in supervised learning and the clustering algorithm in unsupervised learning can effectively classify the data according to the special properties and characteristics, and have a good classification effect. For a given data, the methods that can effectively solve the classification problem include K nearest neighbor algorithm and decision tree algorithm in classification algorithm, and DBSCAN algorithm and k-means algorithm in clustering algorithm. In this paper, k-nearest neighbor algorithm is used to solve the manifold binary classification problem by voting the nearest neighbor label of the sample. In addition, I used 10 common clustering algorithms from the sklearn package for classification and found that they are not suitable for solving the problem without modification.

Keywords:K Nearest Neighbor Algorithm, Manifold Binary Classification

1 Problem Description and Data Description

Looking first at the required data and the given final classification, we find that the geometric distribution of the data is similar to a spiral structure, which has two manifold structures. Such geometric distribution will interfere with the ordinary classification algorithm, so as to not get the expected classification effect.

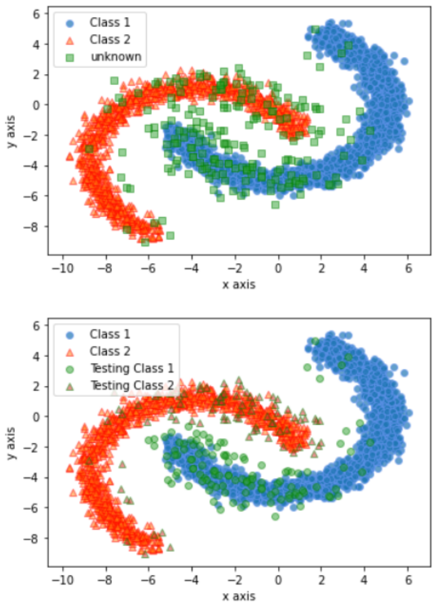


Figure 1: Given Final Classification

As for training data samples, the file *TrainingData.csv* provides 3000 2-dimensional training data samples, and the file *TrainingLabels.csv* provides class label values of training data samples, with 1 and 2 values, representing negative and positive respectively. About the test data samples, the file *TestingData.csv* gives 200 two-dimensional data, and the class labels of the test data samples are given by the file *TestingLabels.csv*, which can be used to calculate the confusion matrix, accuracy, sensitivity, specificity, accuracy and other indicators.

It can be drawn from the graphs that the geometric center of the data is roughly (-2, -2), and the number of points of positive and negative types is roughly the same. Before the training, we used the mean value and standard deviation to standardize the data, and finally the training set, verification set and test set all obeyed the normal distribution with the mean value of 0 and the variance of 1.

2 Introduction to K Nearest Neighbor Algorithm

KNN (K-Nearest Neighbor) algorithm is one of the most basic algorithms in machine learning, which can be used for classification as well as regression. The idea of KNN algorithm is to measure the distance between different eigenvalues, and the category of unmarked samples is decided by the vote of the nearest K neighbors. In k-dimensional space, the distance between different eigenvalues of K nearest neighbor algorithm can be calculated in these following ways:

2.1 Euclidean Distance

Euclidean distance is the most intuitively understood distance measurement method. It is calculated by the arithmetic square root of the sum of squares of distances between two points in various dimensions, which can be calculated by the following formula:

$$D_{i,j} = \sqrt{\sum_{\mathrm{k=1}}^{\mathrm{p}} (\mathrm{x_{ki}} - \mathrm{x_{kj}})^2}$$

2.2 Manhattan Distance

Manhattan is the sum of the absolute value of the distance between two points in each dimension, which can be calculated by the following formula:

$$D_{i,j} = \sum_{k=1}^{p} |x_{ki} - x_{kj}|$$

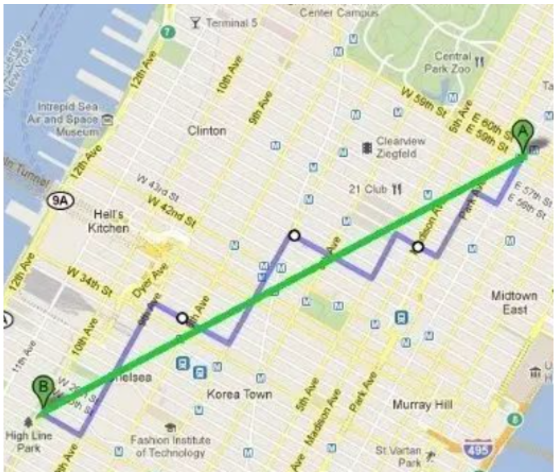


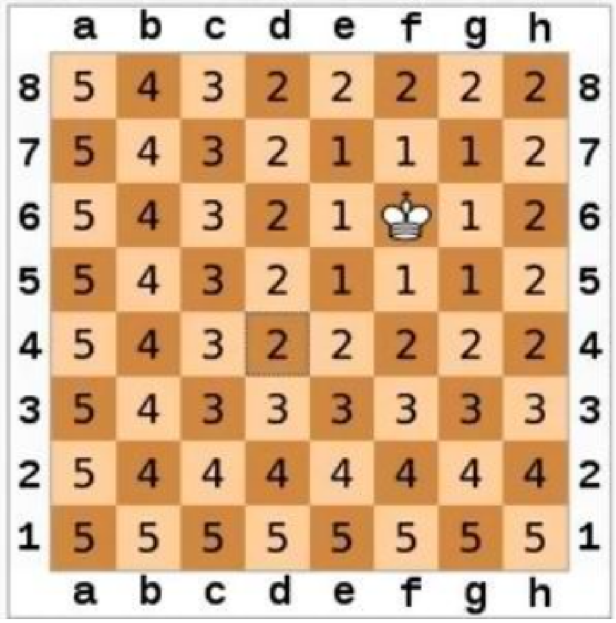
Figure 2: Manhattan Distance Compared with Euclidean Distance

We use concrete images to represent the differences between these 2 distance measures. From the map of downtown Manhattan, The Euclidean distance is compared with the distance of Manhattan in two-dimensional space. We can see that the Euclidean distance between two points is its straight line distance, while the distance between Manhattan is its broken line distance.

2.3 Chebyshev Distance

Chebyshev Distance is the maximum distance between the dimensions of two points, which can be calculated by the following formula:

$$D_{i,j} = \max_{k=1} \left(|x_{ki} - x_{kj}| \right)$$



	a	b	c	d	e	f	g	h	
8	5	4	3	2	2	2	2	2	8
7	5	4	3	2	1	1	1	2	7
6	5	4	3	2	1	♔	1	2	6
5	5	4	3	2	1	1	1	2	5
4	5	4	3	2	2	2	2	2	4
3	5	4	3	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4	2
1	5	5	5	5	5	5	5	5	1
	a	b	c	d	e	f	g	h	

Figure 3: Chebyshev Distance

Chebyshev distance is often explained by the actions of a chess king. In chess, the king has eight equidistant options, and this distance is the Chebyshev distance.

In this way, the K-nearest algorithm will select the nearest K neighbors according to the number of nearest neighbors, and vote according to the categories of these K neighbors, so as to obtain the category of the sample.

For example, if most of the K neighbors are of type 1(Negative), the probability of this point's type is 1(Negative).

3 Model Construction

When machine learning algorithm is used for prediction, K-fold cross-validation is often used to improve the accuracy of prediction. The so-called K-fold cross-validation is to divide the data set into k groups by some sampling method, in which the k-1 groups serve as the training set and the remaining one serves as the verification set. After the training set is used to train the classifier, the verification set is used to test the trained model. Each group was tested once, and the final prediction result was obtained by averaging the results of the k tests. This method can get accurate results, among which 10-fold cross-validation is the most commonly used method. Here we cross-validate by using the *StratifiedKFold* function in *sklearn* to sample the data set to 10 subsamples.

In addition to determining the number of training sets, according to the principle of KNN algorithm, we also need to determine the number of neighbors, namely, the value of k.[1]

According to the voting principle, the minority should obey the majority. Theoretically, we should choose an odd number as the value of k. But through the experiments found that even k can also get good classification. Here, we choose 1 to 10 as the value of k, and find that the prediction accuracy is perfect no matter under which distance calculation method. The predicted results are shown as follows:

Table 1: Prediction Accuracy Results

n_neighbors	cal_method	accuracy	n_neighbors	cal_method	accuracy
1	educlidean	1.0	6	educlidean	1.0
1	manhattan	1.0	6	manhattan	1.0
1	chebyshev	1.0	6	chebyshev	1.0
2	educlidean	1.0	7	educlidean	1.0
2	manhattan	1.0	7	manhattan	1.0
2	chebyshev	1.0	7	chebyshev	1.0
3	educlidean	1.0	8	educlidean	1.0
3	manhattan	1.0	8	manhattan	1.0
3	chebyshev	1.0	8	chebyshev	1.0
4	educlidean	1.0	9	educlidean	1.0
4	manhattan	1.0	9	manhattan	1.0
4	chebyshev	1.0	9	chebyshev	1.0
5	educlidean	1.0	10	educlidean	1.0
5	manhattan	1.0	10	manhattan	1.0
5	chebyshev	1.0	10	chebyshev	1.0

[1] k here refers to k of KNN, that is, the number of the nearest points selected, rather than k for k-fold cross-verification.

4 Result Presentation

Through the algorithm construction, we can easily plot the classification prediction of the test set data. We randomly selected three parameter combinations and drew the scatter diagram of classification situation as follows.

4.1 7 Chebyshev

Firstly, the number of neighbors was selected as 7, and then Chebyshev distance was used for distance calculation. The prediction results were obtained by KNN classification, as shown in the figure.

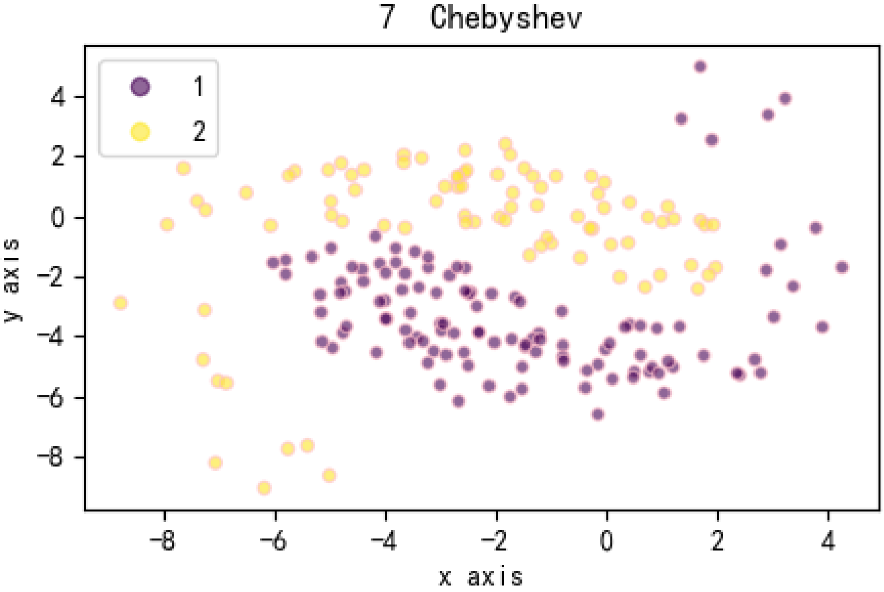


Figure 4: Prediction of 7 Chebyshev

4.2 6 Manhattan

Secondly, the number of neighbors was selected as 6, and then Manhattan distance was used for distance calculation. The prediction results were obtained through KNN classification, as shown in the figure.

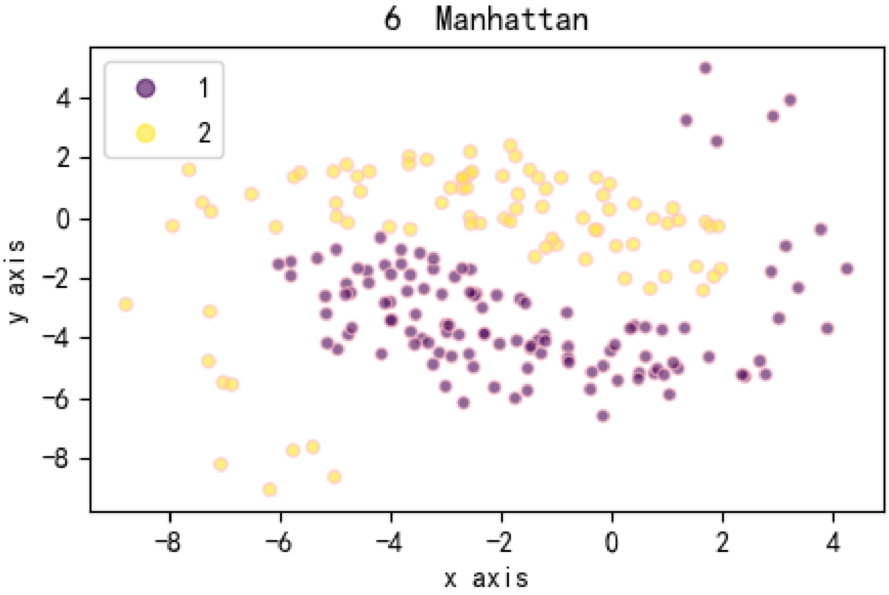


Figure 5: Prediction of 6 Manhattan

4.3 5 Educlidean

Finally, the number of neighbors was selected as 5, and then Euclidean distance was used for distance calculation. The prediction results were obtained through KNN classification, as shown in the figure.

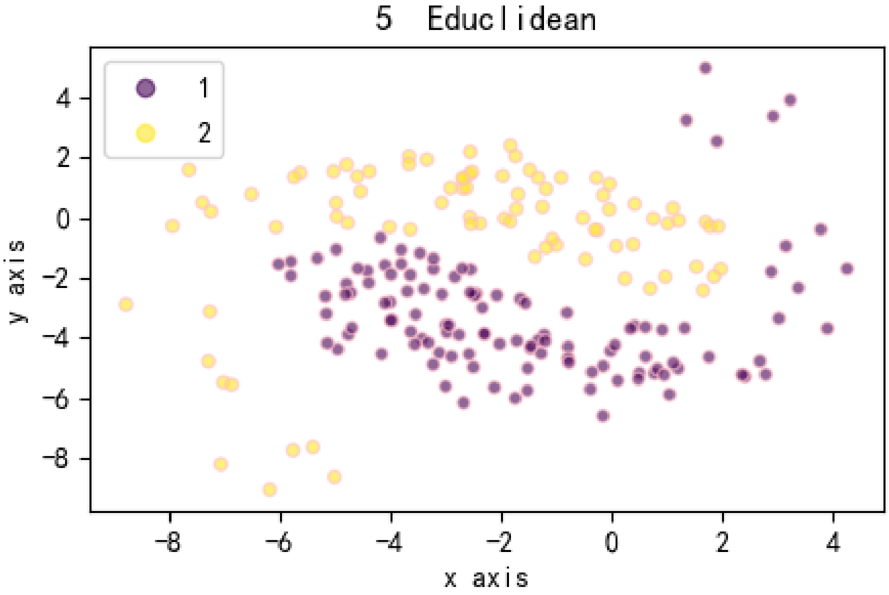


Figure 6: Prediction of 5 Educlidean

4.4 Conclusion

By comparing the results of the three images, it is found that the classification results are identical and completely accurate. The result shows that KNN algorithm can solve this problem perfectly. It seems that neither the selection of the number of neighbors nor the selection of distance calculation method will affect the accuracy of classification prediction. But this argument is remained untested.

At the same time, we can also find that the test set seems to have the same distribution as the training set, and has roughly the same geometric center, namely (-2, -2). Meanwhile, types 1 (negative) and 2 (positive) have roughly the same number of individuals and have similar manifold distributions.

5 Model Verification and Conclusion

I then use a number of indicators to test whether the model's prediction results are superior, including confusion matrix, accuracy, precision, F1, ROC_AUC and so on. Here I choose the latest results above to test.

5.1 Mathematical Interpretation

Before get into Verification Indicators, we should first understand these 4 items, which consist of confusion matrix.

Table 2: Item Description

item	sign	description
True Positive	TP	Prediction is positive and correct
False Positive	FP	Prediction is Positive but wrong
True Negative	TN	Prediction is negative and correct
False Negative	FN	Prediction is negative but wrong

By these 4 items we can calculate confusion matrix and some other verification indicators.

5.2 Verification Indicators

Table 3: Confusion Matrix

Confusion Matrix		Predicted Value	
		Positive	Negtive
Real Value	Positive	116	0
	Negative	0	84

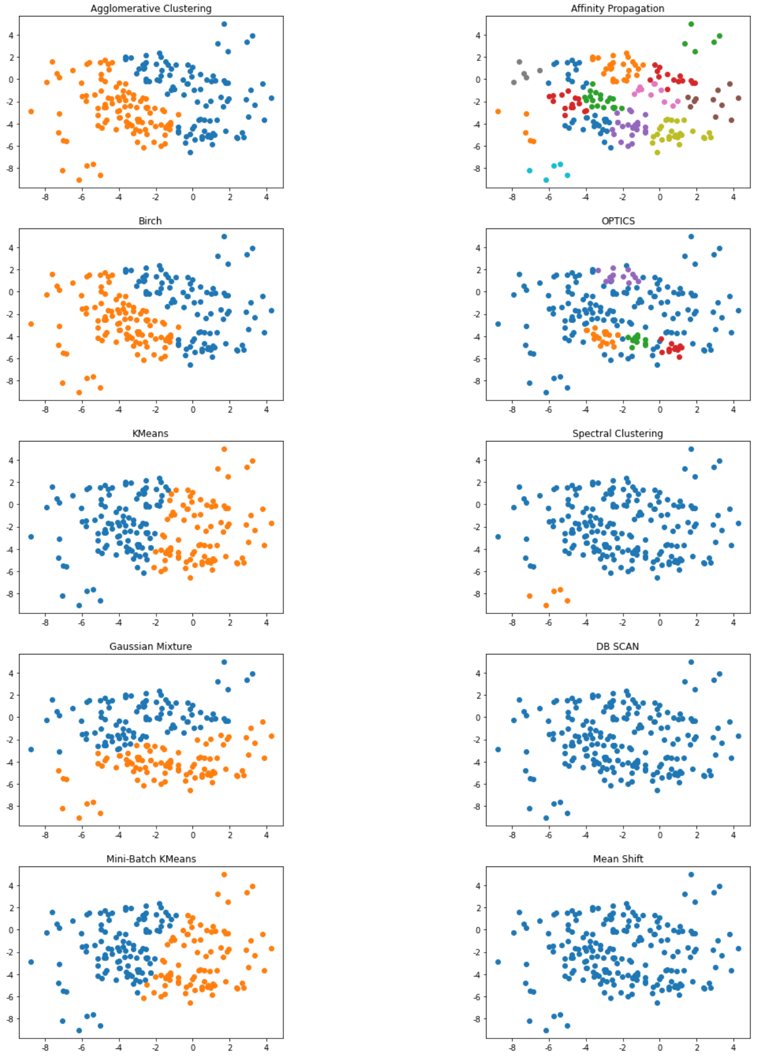
Here, we output precision, recall, F1, and accuracy directly using the classification_report function in the sklearn package.

Table 4: Classification Report

	precision	recall	f1-score	support
1	1.00	1.00	1.00	116
2	1.00	1.00	1.00	84
accuracy			1.00	200
macro avg	1.00	1.00	1.00	200
weighted avg	1.00	1.00	1.00	200

6 Clustering Algorithm

In addition, I used 10 common clustering algorithms from the sklearn package for classification. They are: Agglomerative Clustering, Birch, KMeans, Gaussian Mixture, Mini-Batch KMeans, Affinity Propagation, OPTICS, Spectral Clustering, DBSCAN and Mean Shift.



(a) subfig1

(b) subfig2

Figure 7: Results of Clustering Algorithm

These 10 algorithms and their results are presented ablove. Obviously, they fail confronting mainfold binary classification problem by applying mechanically.

7 Conclusion

Obviously, this KNN model is robust and superior confronting manifold binary classification problem. Nonparametric models are often more suitable for solving manifold classification problems, and K-nearest neighbor algorithm performs better in this task, depending on the principle of k-nearest neighbor, that is, finding the label of the nearest point.

However, if we directly use the clustering algorithms from sklearn package without modification, this problem would fail to be solved. This result is foreseeable, which indicates that we have to analyse problems case by case rather than simply apply former algorithms mechanically.