Comparison Classification Prediction of Iris Class using KNN, Linear Regression and Logistic Regression

Yiwei Shu

University of Illinois at Urbana-Champaign
IL, US
yiweis3@illinois.edu

***Abstract*—Cluster prediction is an important task in machine learning that involves grouping similar data points together. In this paper, we compare the performance of three popular machine learning algorithms - KNN, Linear Regression, and Logistic Regression for cluster prediction of the iris class. By using the iris dataset and evaluating the algorithms to predict the iris class, our result shows that KNN and outperforms both Linear Regression and Logistic Regression in terms of accuracy score.**

***Index Terms*—Classification, KNN, Linear Regression, Logistic Regression**

I. Introduction

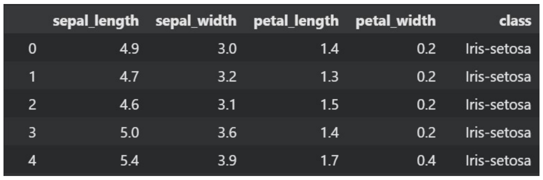
The iris dataset is a well-known dataset in machine learning, which is always used for classification tasks. It consists of 150 samples, each with five coloumns: sepal length, sepal width, petal length, petal width and the iris class, which is the response variable. In this paper, we focus on the cluster prediction of the iris class using three popular machine learning algorithms - KNN, Linear Regression, and Logistic Regression.

II. Raw Data Analysis and Preprocessing

A. Analysis of the Correlation

The original data have 150 rows and 5 columns, in which there are four features: sepal length, sepal width, petal length, and petal width and ine response variable: class. The Fig. 1 shows first five datas of the total dataset. We define:

class	class number
Iris-setosa	1
Iris-versicolor	2
Iris-virginica	3



	sepal_length	sepal_width	petal_length	petal_width	class
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

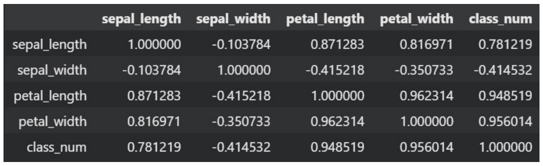
Fig. 1. First five datas of the total dataset.

By calculating the correlation, we can find the most correlated features from them. As the Fig. 2 shows, the petal size (petal length and petal width) shows further higher correlation than the sepal size.

$$r_{petal-length} = 0.948519 > r_{sepal-length} = 0.781219$$

$$r_{petal-width} = 0.956014 > |r_{sepal-width}| = 0.414532$$

Thus, we use the feature petal length and petal width to predict the class of iris.



	sepal_length	sepal_width	petal_length	petal_width	class_num
sepal_length	1.000000	-0.103784	0.871283	0.816971	0.781219
sepal_width	-0.103784	1.000000	-0.415218	-0.350733	-0.414532
petal_length	0.871283	-0.415218	1.000000	0.962314	0.948519
petal_width	0.816971	-0.350733	0.962314	1.000000	0.956014
class_num	0.781219	-0.414532	0.948519	0.956014	1.000000

Fig. 2. Correlation of the features and the response variable.

With feature of petal length and petal width, we can find the iris dataset shows clear classification distribution trend. Like Fig. 3 shows.

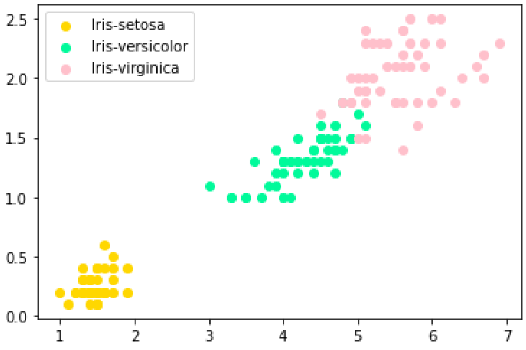


Fig. 3. Distribution of iris dataset.

B. Train data and test data preprocessing

We use $\frac{3}{4}$ of the total data to train the model, and use the left to test the model so that we can find the accuracy. All of the choices are total random. Then we standardize all data to reduce errors.

III. Machine Learning Algorithms Train and Test

A. K-Nearest Neighbors (KNN) Algorithms

K-Nearest Neighbors (KNN) is a simple machine learning algorithm used for classification and regression tasks. In KNN, the prediction for a new instance is based on the K-nearest neighbors, i.e., the K closest instances to the new instance in the training set. (source: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm) It is always used in predicting of the classification.

$$d(x,y) = \sqrt{((x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2)}$$

Calculate the distance between the new instance and all instances in the training set using a distance metric. Then select the K instances with the shortest distance to the new instance. And then we can determine the class label of the new instance by taking the majority vote of the K-nearest neighbors.

In our prediction, we use the package imported from sklearn and train the data.

```
from sklearn.neighbors import
   KNeighborsClassifier
model1 = KNeighborsClassifier()
model1.fit(X_train,y_train)
predictions1 = model1.predict(X_test)
```

We use accuracy score to judge the mode.

$$AccuracyScore = \frac{TP + TN}{TP + TN + FP + FN}$$

The accuracy score outcome is 1.0, which means there are no false prediction for the test data.

The Fig. 4 shows the outcome of KNN. (Green dots are true prediction, Reds dots are false one.)

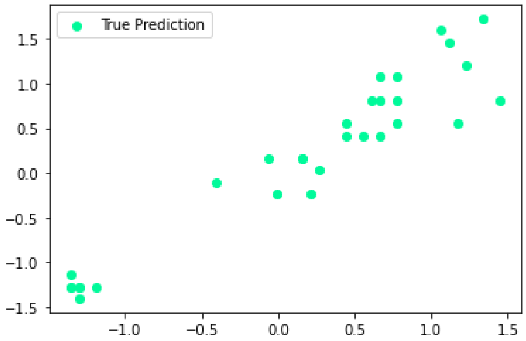


Fig. 4. Outcome of KNN.

B. Logistic Regression Algorithms

Logistic Regression is a popular classification algorithm used in machine learning to predict the probability of a binary or multiclass outcome. It is a type of supervised learning algorithm that is used when the response variable is categorical in nature. (source: https://en.wikipedia.org/wiki/Logistic_regression) The Logistic Regression algorithm works by modeling the probability of the outcome variable as a function of the predictor variables. It uses a logistic function to transform the output of a linear regression into a probability value between 0 and 1. It can be seen as a broader linear regression. The logistic function is defined as:

$$p(x) = \frac{1}{(1 + e^{-z})}$$

$$z = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + ... + b_n \cdot x_n$$

b_0 is the intercept, $b_1, b_2, ..., b_n$ are the coefficients of the predictor variables, and $x_1, x_2, ..., x_n$are the predictor variables.

In our prediction, we use the package imported from sklearn and train the data.

```
from sklearn.linear_model import
   LogisticRegression
model2 = LogisticRegression()
model2.fit(X_train, y_train)
predictions2 = model2.predict(X_test)
```

The accuracy score outcome is 0.9. The Fig. 5 shows the outcome of Logistic Regression.

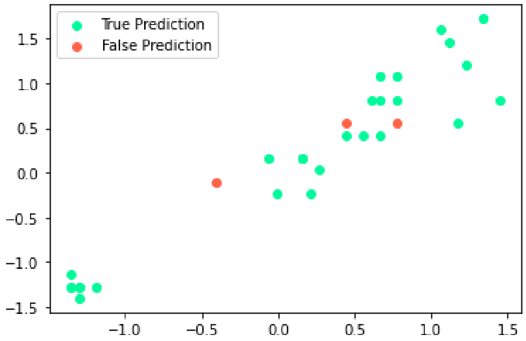


Fig. 5. Outcome of Logistic Regression.

C. Linear Regression Algorithms

Linear Regression is a popular algorithm used in machine learning to model the relationship between a dependent variable and one or more independent variables. It is a type of supervised learning algorithm that is used when the response variable is continuous in nature. (source: https://en.wikipedia.org/wiki/Linear_regression) The Linear Regression algorithm works by finding the best-fit line that describes the linear relationship between the dependent variable and the independent variables. The best-fit line is the one that minimizes the sum of the squared errors between the predicted values and the actual values of the dependent variable.

The linear regression equation can be written as:

$$y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + ... + b_n \cdot x_n$$

For our iris dataset, it is essentially a classification problem. However, I want to try to connect the classification and regression in this problem by setting range for the outcome of the regression outcomes so that it can be usable in classification. We define:

Regression Outcome	Classification
$0 \leq$ y ≤ 1.4	1
$1.5 \leq$ y ≤ 2.4	2
$2.5 \leq$ y ≤ 4	3

In our prediction, we use the package imported from sklearn and train the data.

```
from sklearn.linear_model import
   LinearRegression
model3 = LinearRegression()
model3.fit(X_train, y_train)
predictions3 = model3.predict(X_test)
```

The accuracy score outcome is 1.0, which means there are no false prediction for the test data. The Fig. 6 shows the outcome of Linear Regression.

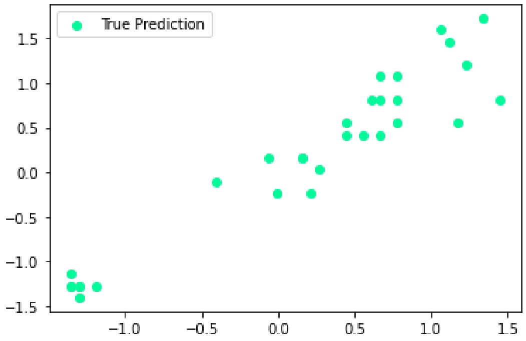


Fig. 6. Outcome of Linear Regression.

IV. Conclusion

Our results show that under the iris dataset, KNN and Linear Regression outperforms Logistic Regression in terms of accuracy . Both of KNN and Linear Regression achieved an accuracy of 1, while Linear Regression and Logistic Regression achieved accuracies of 0.9. However, for the iris dataset, the number of test are just 30. The number if test data is not large enough. If there are more data, the accuracies of KNN and Linear Regression maybe show something different.

Besides, for the method of connecting the regression and the classification during deal with the classification by using the linear regression has some limits. In the situation of iris dataset, the raw data of three class shows large differences on the petal length and petal width. It make the classification easier to distinguish. But for the case of more complict dataset, it maybe not proper.

Acknowledgment

As an exchange student, STAT107 is an attractive course. I gained fun and knowledge from this term course. I would like to express my appreciation to professor Karle and professor Wade. You make the course full of enthusiasm. And I would like to express my appreciation to TAs and CAs, who always help me in time.

References

[1] Alpaydin, E. (2010). Introduction to machine learning (2nd ed.). MIT Press.

[2] Chandrasekaran, R., Srinivasan, D. (2015). Prediction of plant height using linear regression model: A case study on Iris dataset. International Journal of Advanced Research in Computer Science and Software Engineering, 5(7), 166-170.

[3] Chuan, C., Li, Y. (2014). Application of logistic regression analysis in iris classification. Procedia Computer Science, 31, 906-913.