# Introduction

K-Nearest Neighbors is a classification algorithm that can be used in machine learning. It is a method for computer to make prediction for some uncategorized data based on the data with known categories. Add a new data with unknown category. And classify the new cell by looking at the nearest annotated cells. If the ‘K’ in ‘K-nearest neighbors’ is equal to 1, the new data will belong to its nearest neighbor’s category. If the new data are on the halfway between two or more categories, we set K=11 or K=21 to pick the category that get the most votes.

In this project we apply this algorithm to 70000 instances of 10 clothing pictures, each of which is identified with a numeric label (0 through 9). 60000 are training set and 10000 are testing set. After applying this algorithm to training set, we put the whole testing set pixel data into this training algorithm to get predict values for each instance. Then the original label from testing set are set as true values. Finally we compare these two kinds of values using confusion matrix to evaluate the accuracy of this learning algorithm applying to this dataset.

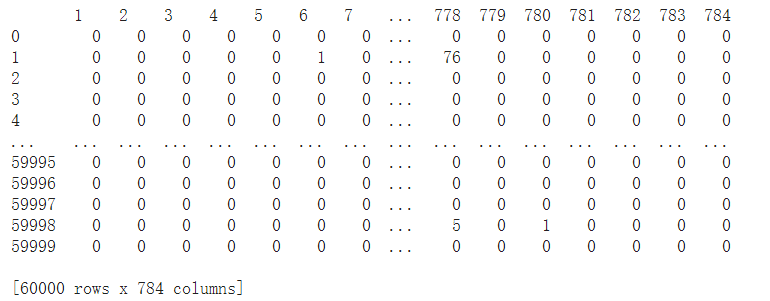
# Goal

* Check and evaluate the accuracy of KNN to our original dataset
* Implement KNN in python, use confusion matrix to find the volume of incorrect prediction between predicted value and true value
* Compare the results of different K-value to pick up the result with better K

# Work

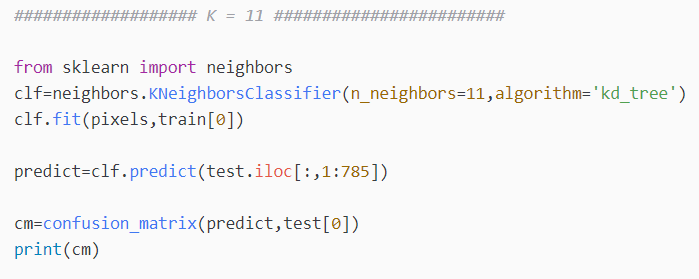


Here is the following dataset that will be applied into the training algorithm:

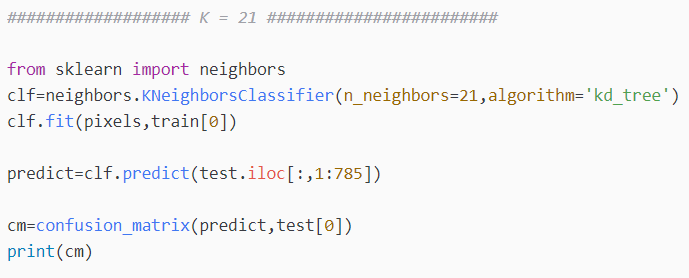


When K = 1, it means that every row of pixels in this dataset will be applied to find the single nearest neighbor. And the label of each row will be set as the same with that nearest neighbor.



When K = 11, it means every row of pixels in this dataset will be applied to find the 11 nearest instances. And the label of each row will be set as the same with the most frequently appeared categories among these 11 neighbors. 

When K = 21, it means every row of pixels in this dataset will be applied to find the 21 nearest instances. And the label of each row will be set as the same with the most frequently appeared categories among these 21 neighbors.

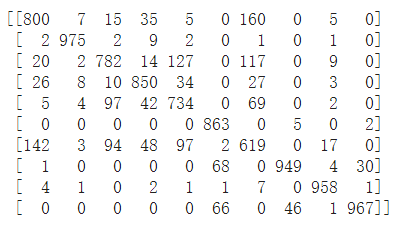


# Findings

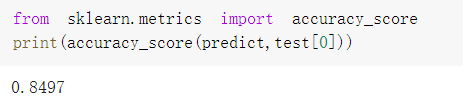
Based on the three training model with different K-values(1,11,21), we conduct 3 confusion matrix(‘cm’ in code) to explain the details of training result.

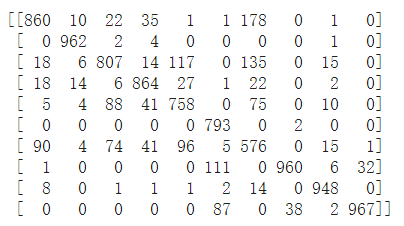
When the accuracy of the model reaches 100%, the diagonal value of the fuzzy matrix is greater than 1, and the rest positions are all zero. This indicates that by using KNN algorithm to train the training set and predict the test set, the predicted value is 100% consistent with the real value. Faced with the pixel data in the training set, the predicted classification of the algorithm is completely consistent with the actual classification. In this case, the model has the best prediction of the real situation, and the model can be directly involved in solving the real image classification problem without labels.

But most of the time, the model's predictions are not entirely accurate. Because of the information covered by the simulation results of the algorithm, it is difficult to cover the data information not seen by the algorithm itself in the full scope.

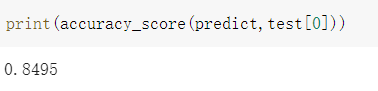


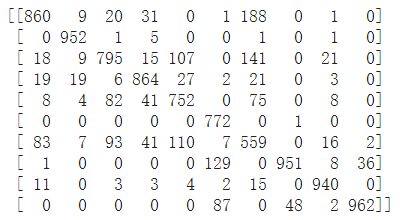
K = 1 confusion matrix



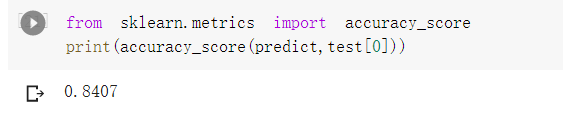


K = 11 confusion matrix





K = 21 confusion matrix



Most of the three predictions fall on the diagonal. This shows that the model has some explanatory power and can give correct prediction in most cases. It is necessary to get a more accurate description of the prediction ability by calculating the accuracy score.

The calculation of accuracy score can be done automatically with Python statements. The idea behind this is to compare the existing matrix with one with a prediction accuracy of 100%. This index can be used to judge the accuracy of the result.

The accuracy score of the training model with K = 1 is 0.8497

The accuracy score of the training model with K = 11 is 0.8495

The accuracy score of the training model with K = 21 is 0.8407

The accuracy of the K = 1 model is better than that of K = 11 and K = 21 when faced with a large amount of test data with a testing set of 10000.