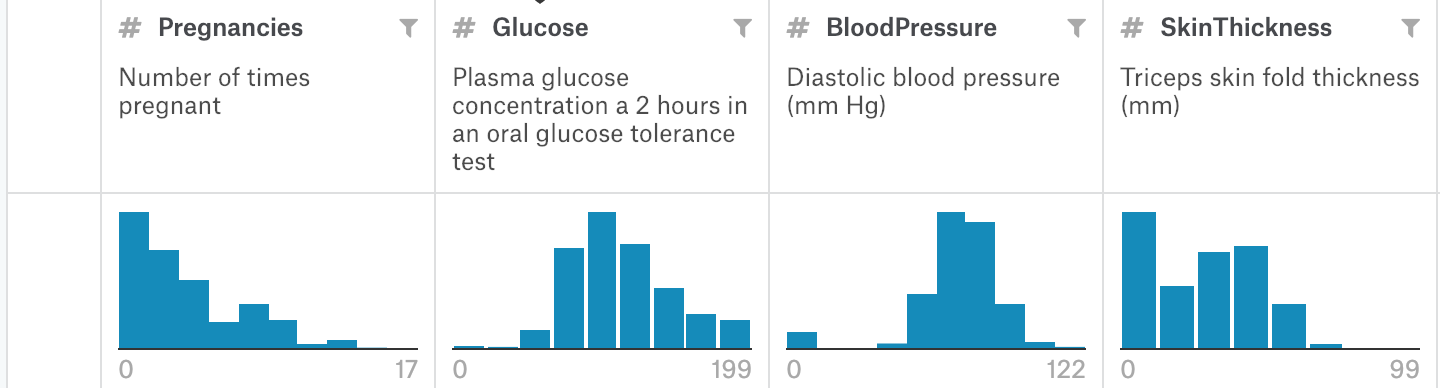
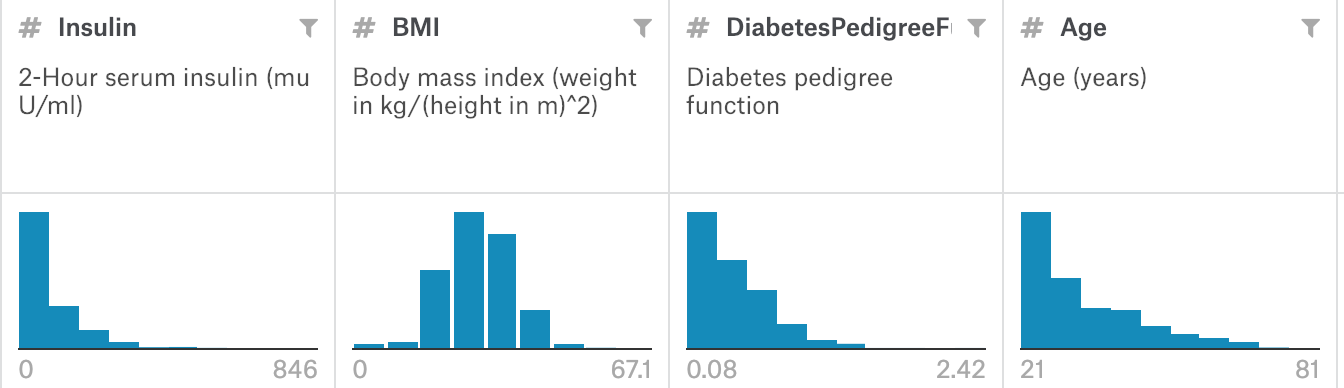
Name:Tzu-Chien, Liu Language: python3.6 Package: sklearn

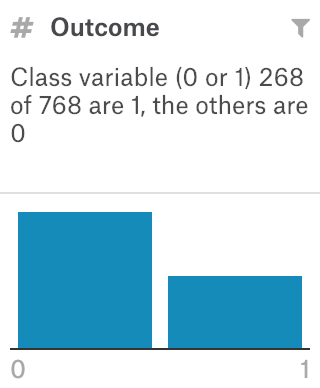
Data Set

Pima Indians Diabetes Database

The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on. The data distribution of each features is listed below:







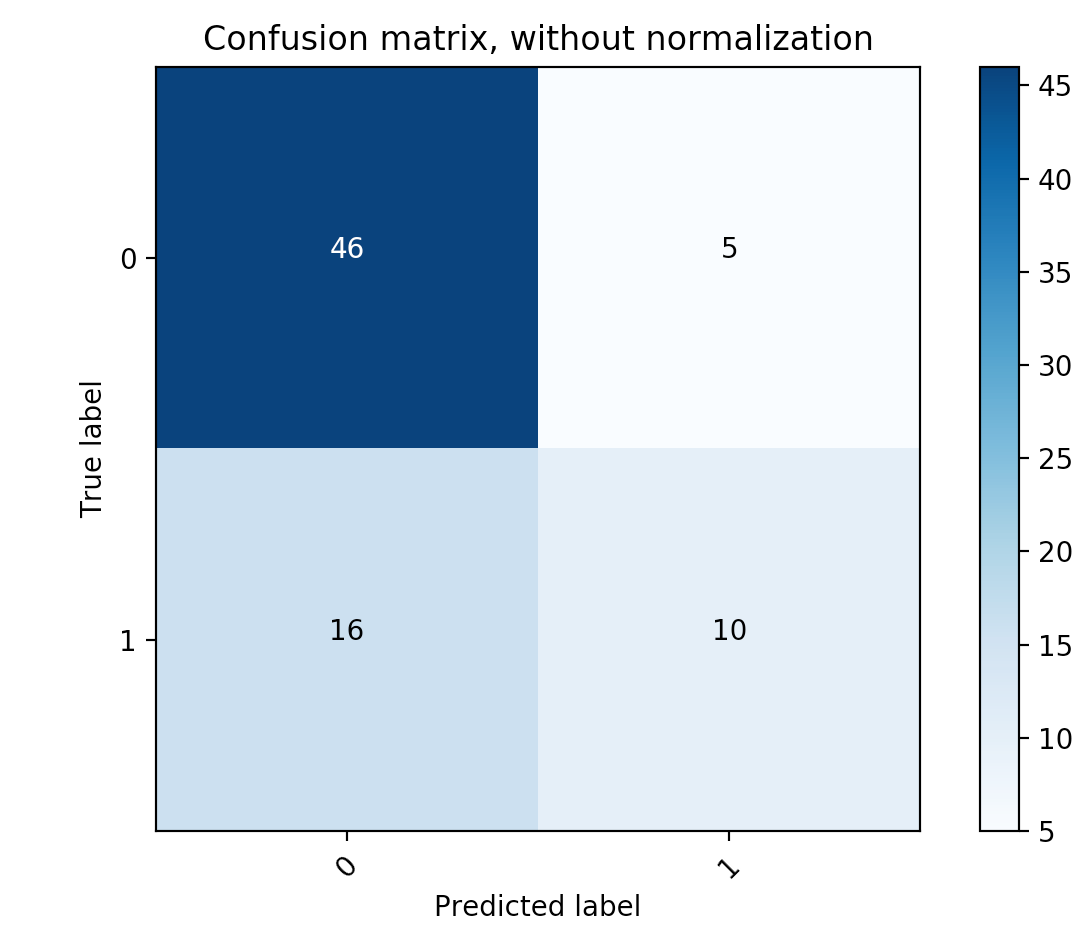
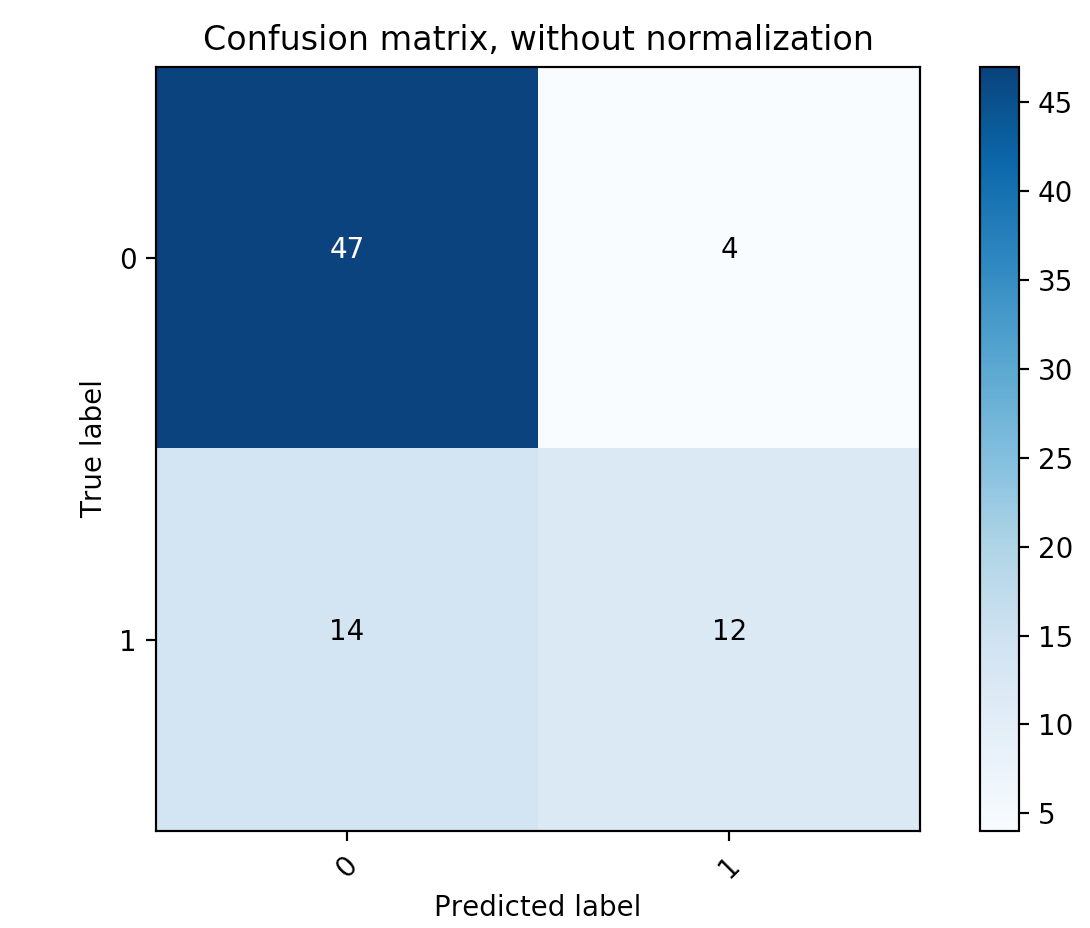
There are 768 data in total. In order to use enough training data, I randomly split data into training data: testing data = 9:1, Thus, there are 691 training data and 77 testing data. In the testing set, the number of the outcome 0 is 59 and the number of the outcome 1 is 18. The proportion of 0 and 1 is still similar to original data. The shortage of this split method is that the testing data is not enough to represent to result of this model.

Algorithm Description:

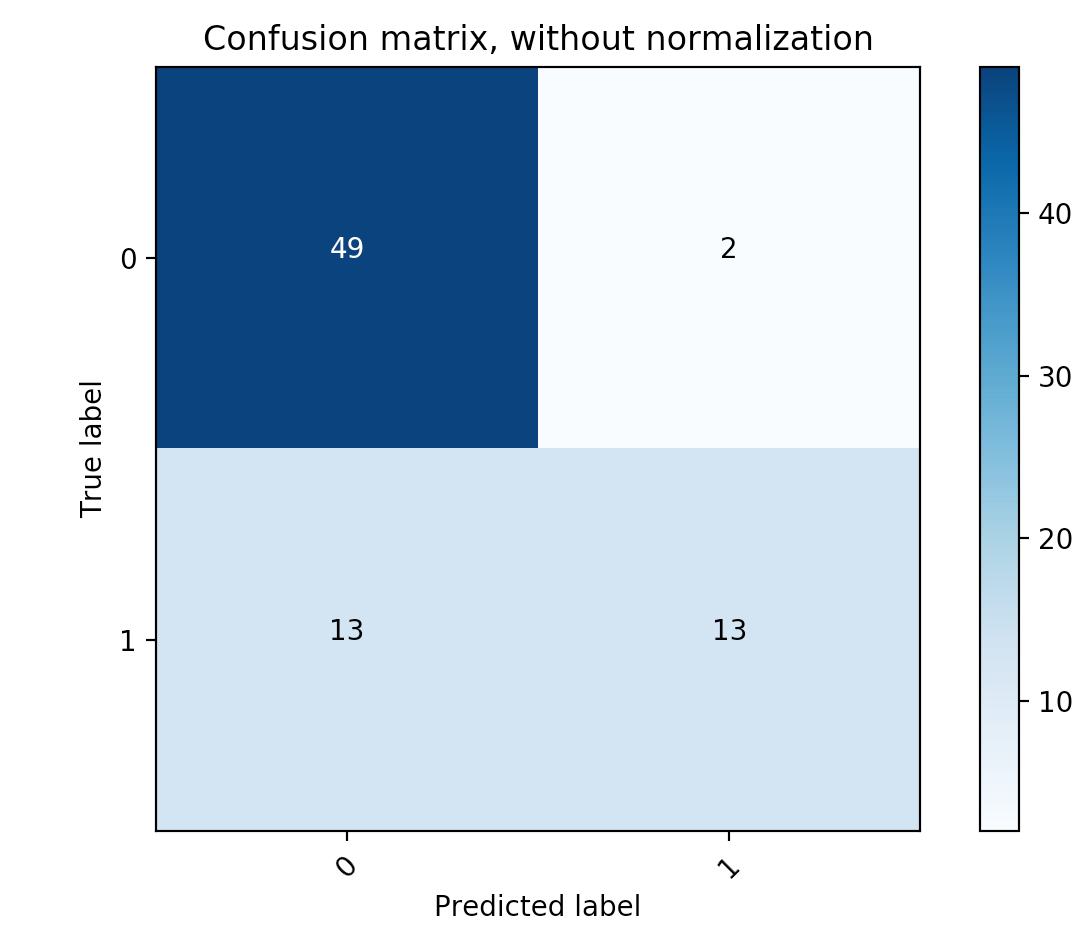
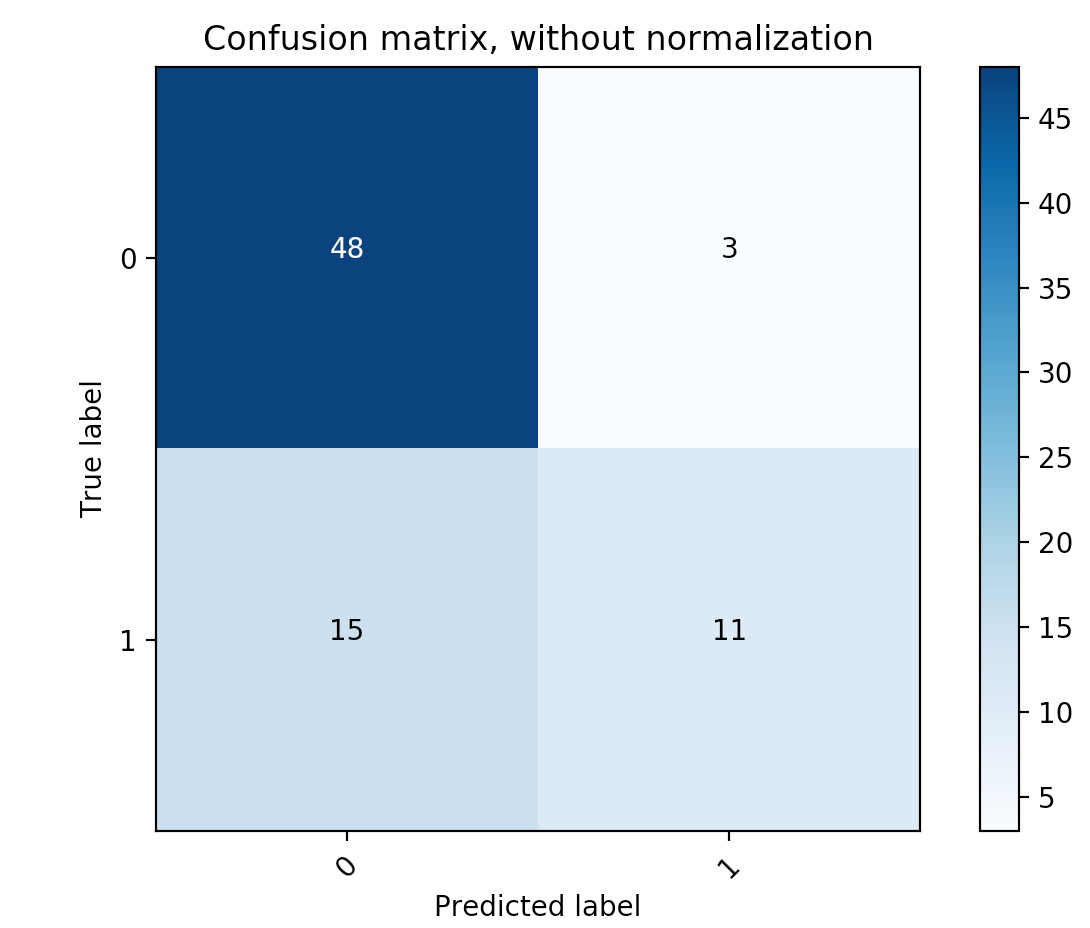
In the data preprocessing stage, I replace the missing data with the mean of the feature data. For the pregnancies feature, I think 0 is also a reasonable number so I don’t deal with pregnancies feature. I remove Insulin feature because too many missing data in this feature which makes it unreasonable to replace missing data with mean value. In the feature scaling stage, I standardize all the features to make the influence of them to model equal. At last, I use KNN as my algorithm.

Algorithm Results:

When K = 3, accuracy :76% When K = 5, accuracy :72.7%



When K = 7, accuracy :76.6% When K = 9 accuracy :80.5%



Runtime:

When K = 3:

training\_time 0.00220s prediction\_time 0.0012669563293457031

When K = 5

training\_time 0.00229s prediction\_time 0.0013401508331298828

When K = 7

training\_time 0.00273s prediction\_time 0.0018620491027832031

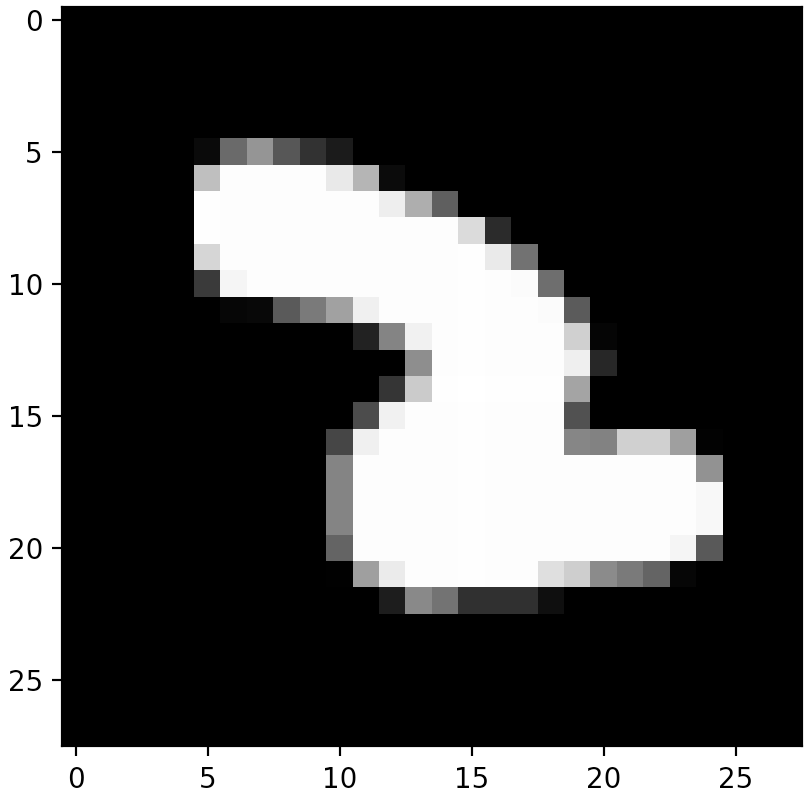
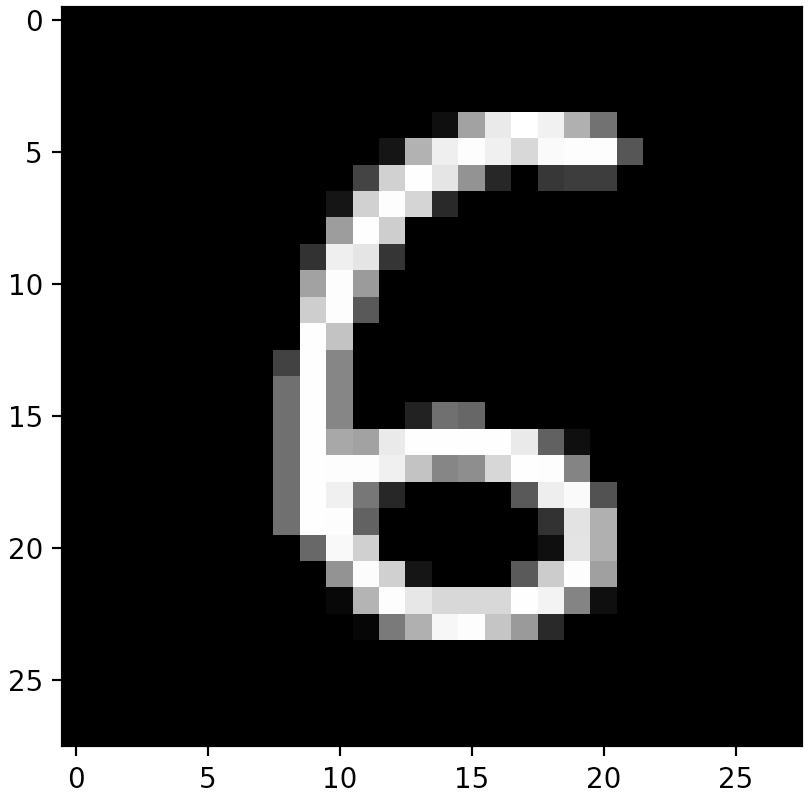
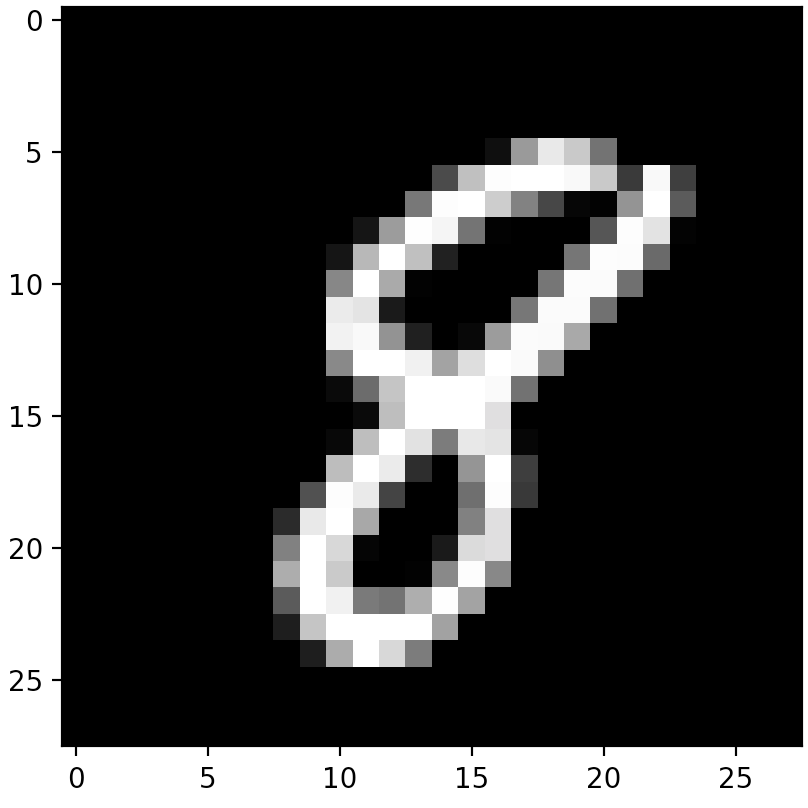
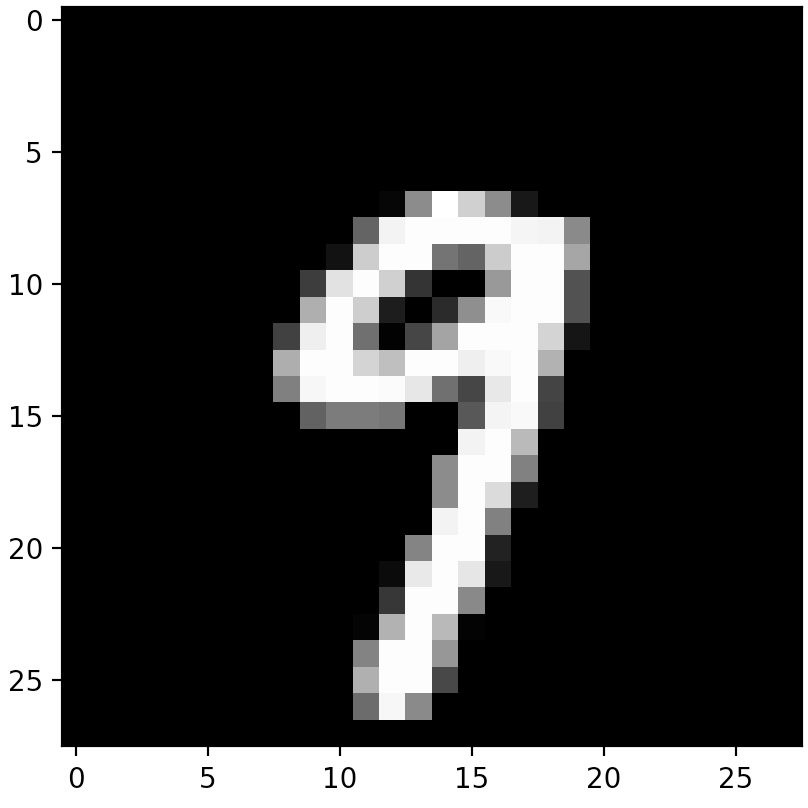
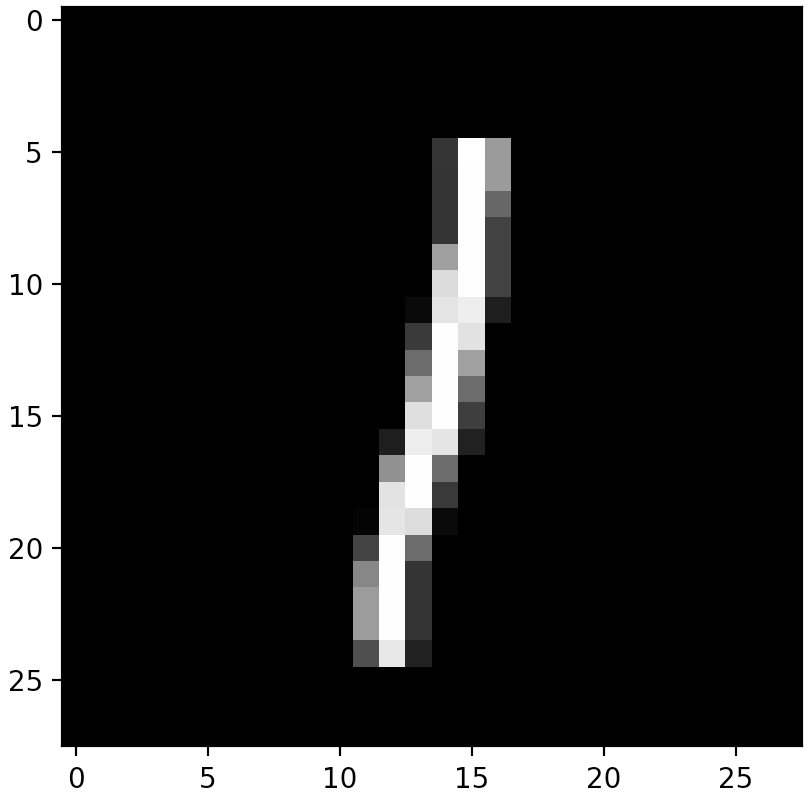
When K = 9

training\_time 0.00293s prediction\_time 0.0019919902801513672

Data Set

minst image database

This dataset consists of images of numbers from 1- 9. Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. The pixel-value is an integer between 0 and 255, inclusive. The image samples are following:



There are totally 42000 images in total. I randomly split 90% of images as training set, 10% of images as testing set. The reason is that training set is already

Algorithm Description:

In the data preprocessing stage, I didn’t do data scaling to this dataset because the pixel values are already on the same scale from 0 to 255. There is no missing data in this dataset. Thus, I directly use each pixel value training feature, use KNN as training model.

Algorithm Results:

In the testing stage, I test total accuracy and select 4 as specific number to analyze.

K = 3, total accuracy:0.967, number 4 accuracy:0.964

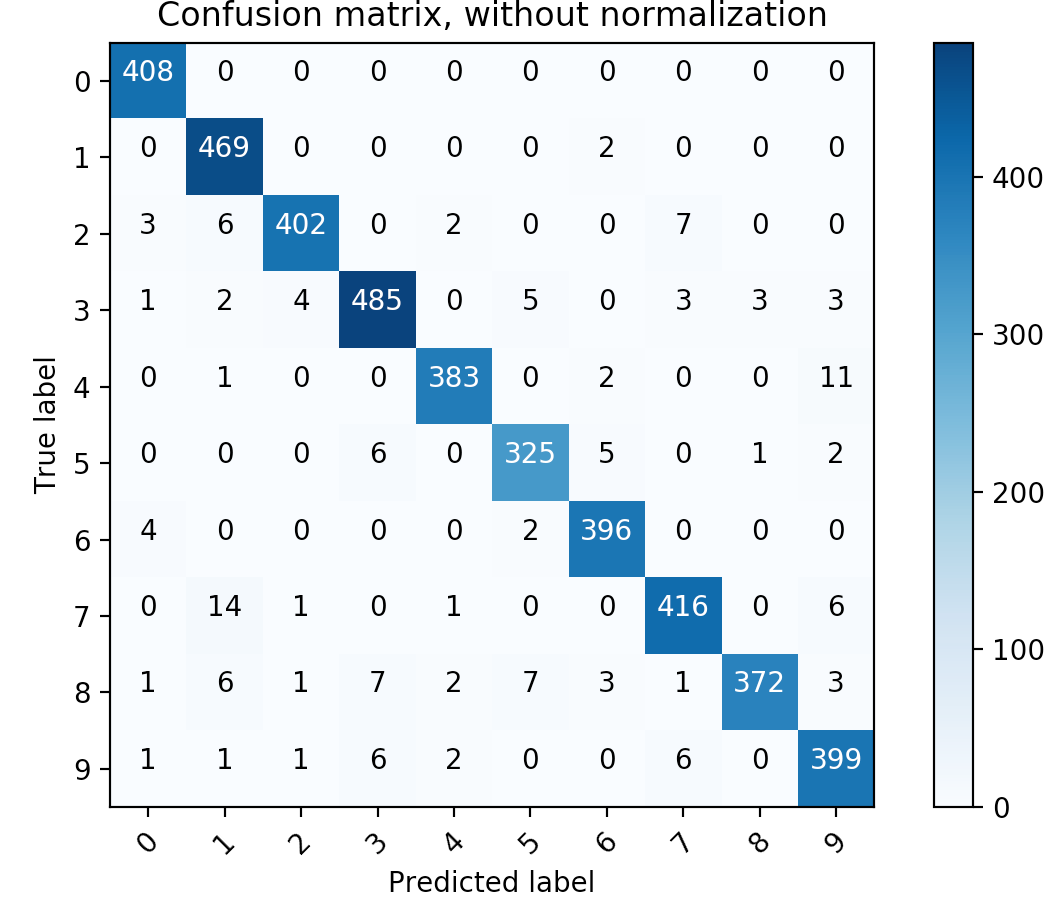
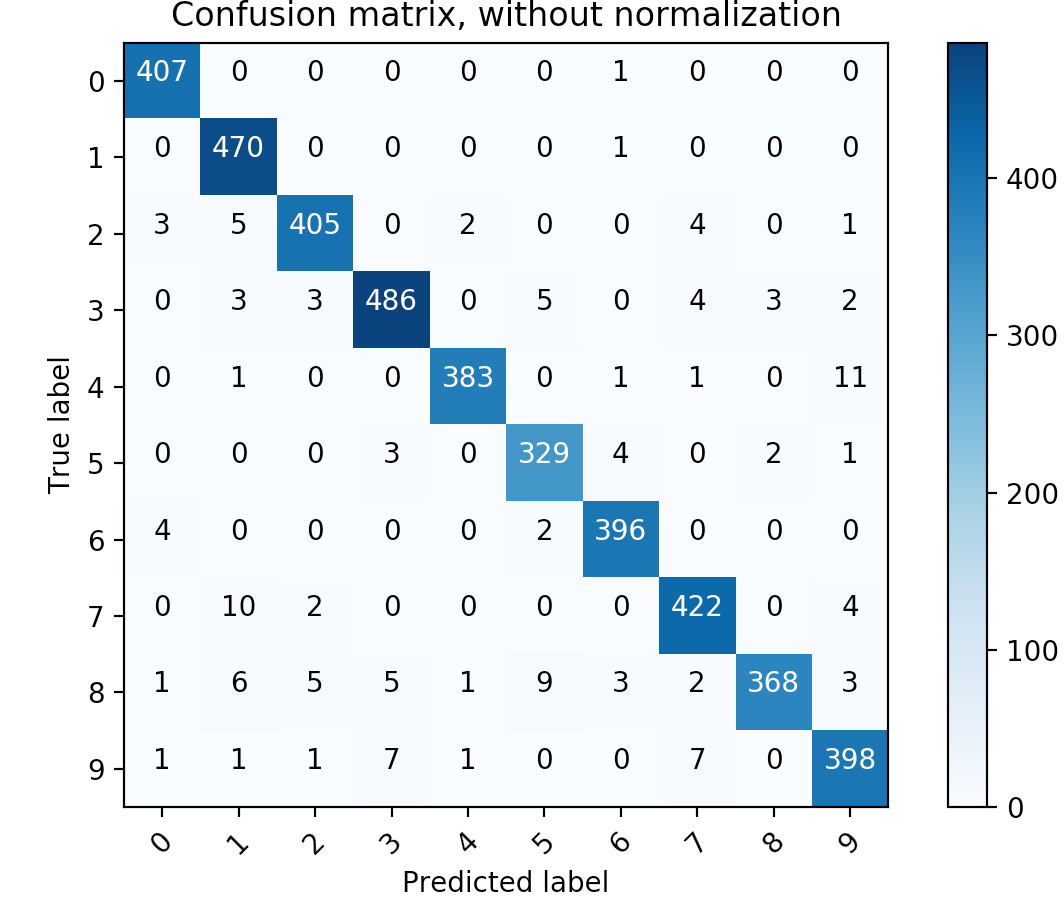
K = 5, total accuracy:0.965, number 4 accuracy:0.964

K = 7, total accuracy:0.964, number 4 accuracy:0.95

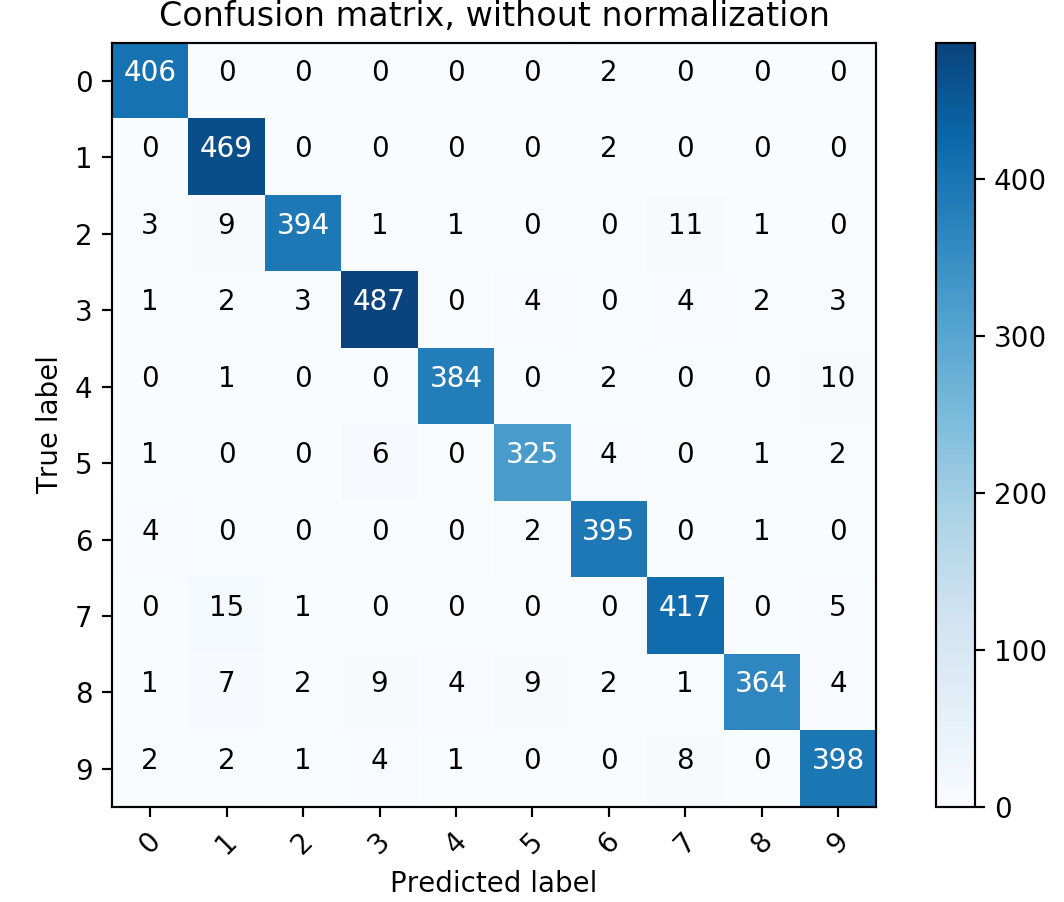
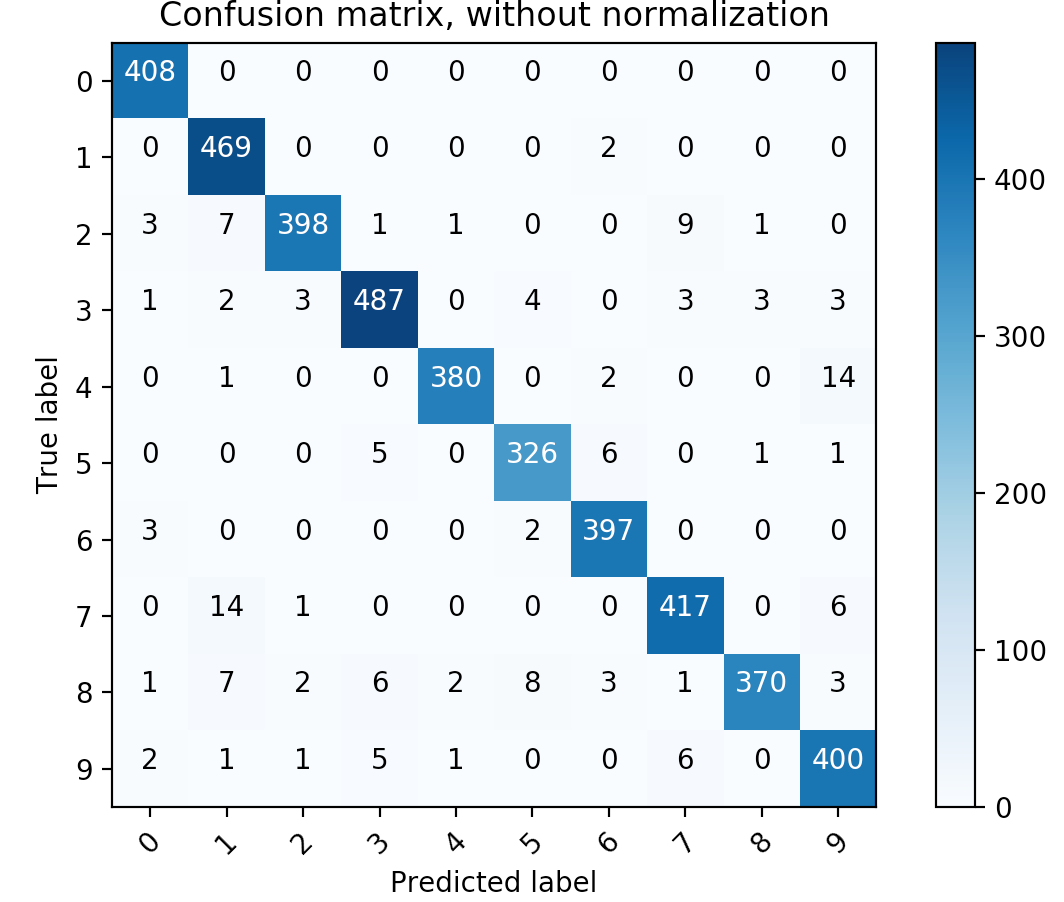
K = 9, total accuracy:0.961, number 4 accuracy:0.967

K= 11, total accuracy:0.958, number 4 accuracy:0.959

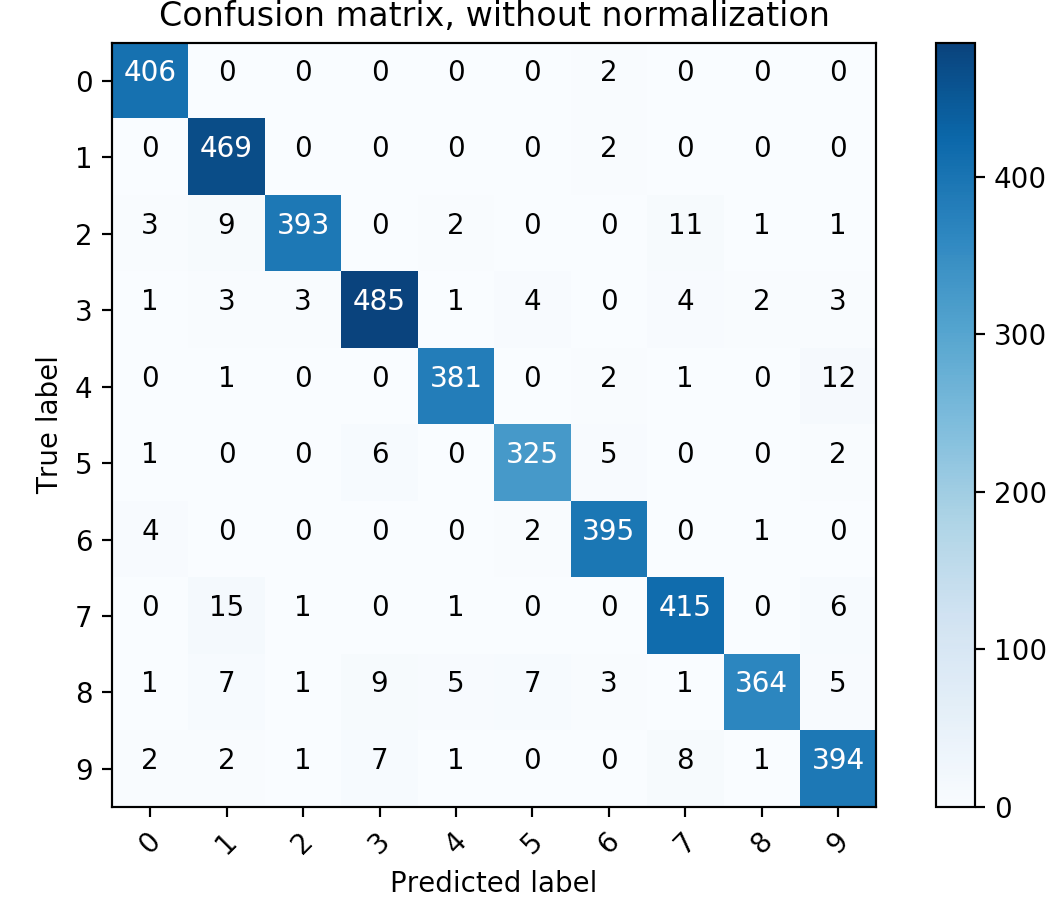
K = 3 K = 5



K = 7 K = 9



K = 11



Accruracy with K

Runtime:

K = 3

Training time: 9.269s Prediction time: 187.308s

K = 5

Training time: 8.9s Prediction time: 186.30856s

K = 7

Training time: 8.6s Prediction time: 183s

K = 9,

Training time: 9.45s Prediction time: 181s

When k= 11

Training time: 8.5s Prediction time: 181s