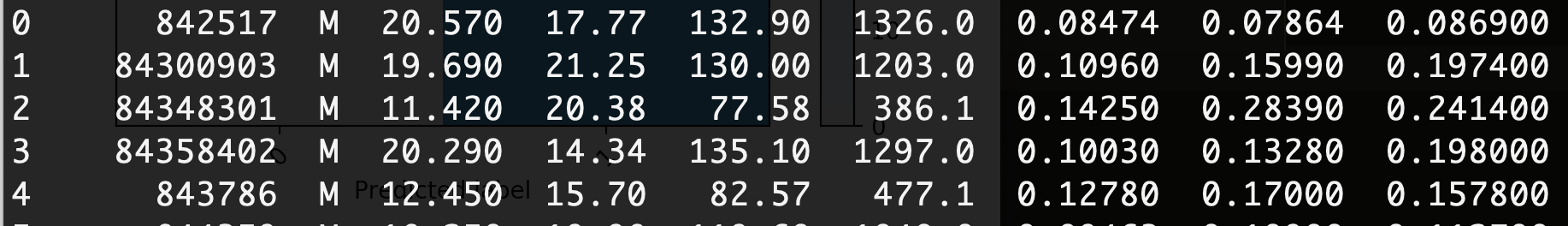
Name:Tzu-Chien, Liu Language: python3.6 Package: Sklearn

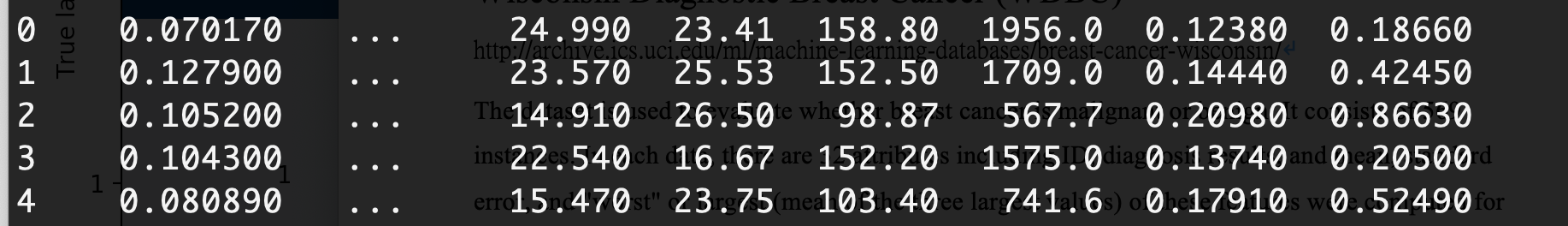
Data Set

Wisconsin Diagnostic Breast Cancer (WDBC)

<http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/>

The dataset is used to evaluate whether breast cancer is malignant or benign. It consists of 569 instances. In each data, there are 32 attributes including ID, diagnosis results, and mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius. There is no missing attributes in the dataset. I randomly split data into training data: testing data = 8:2, Thus, there are 445 training data and 124 testing data. For SVM, 445 is enough for train a good model. In the testing set, the number of the outcome malignant is 68 and the number of the benign 1 is 46. The proportion of benign and malignant is still similar to original data. The result of this split method can properly represent the result of this model. The data example is listed below:



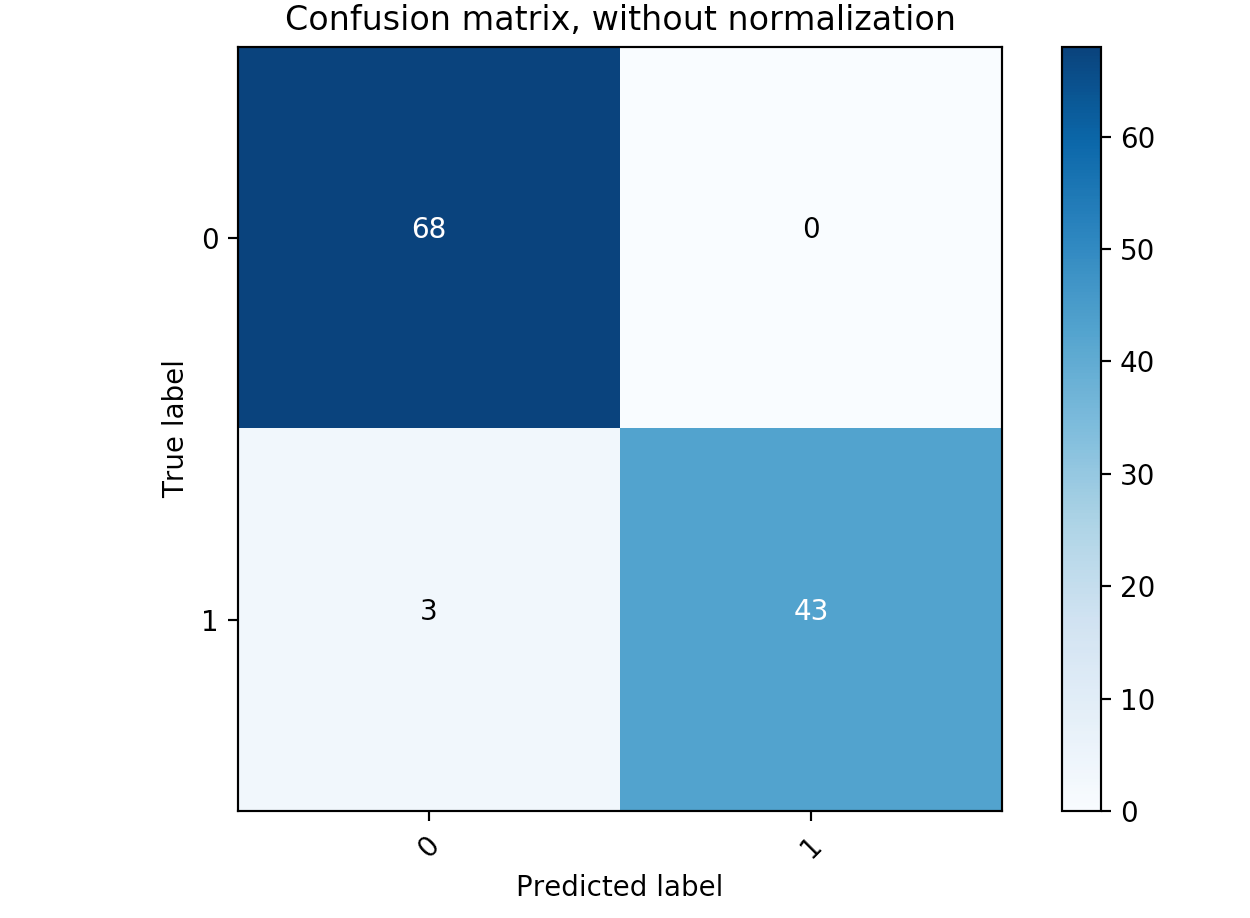
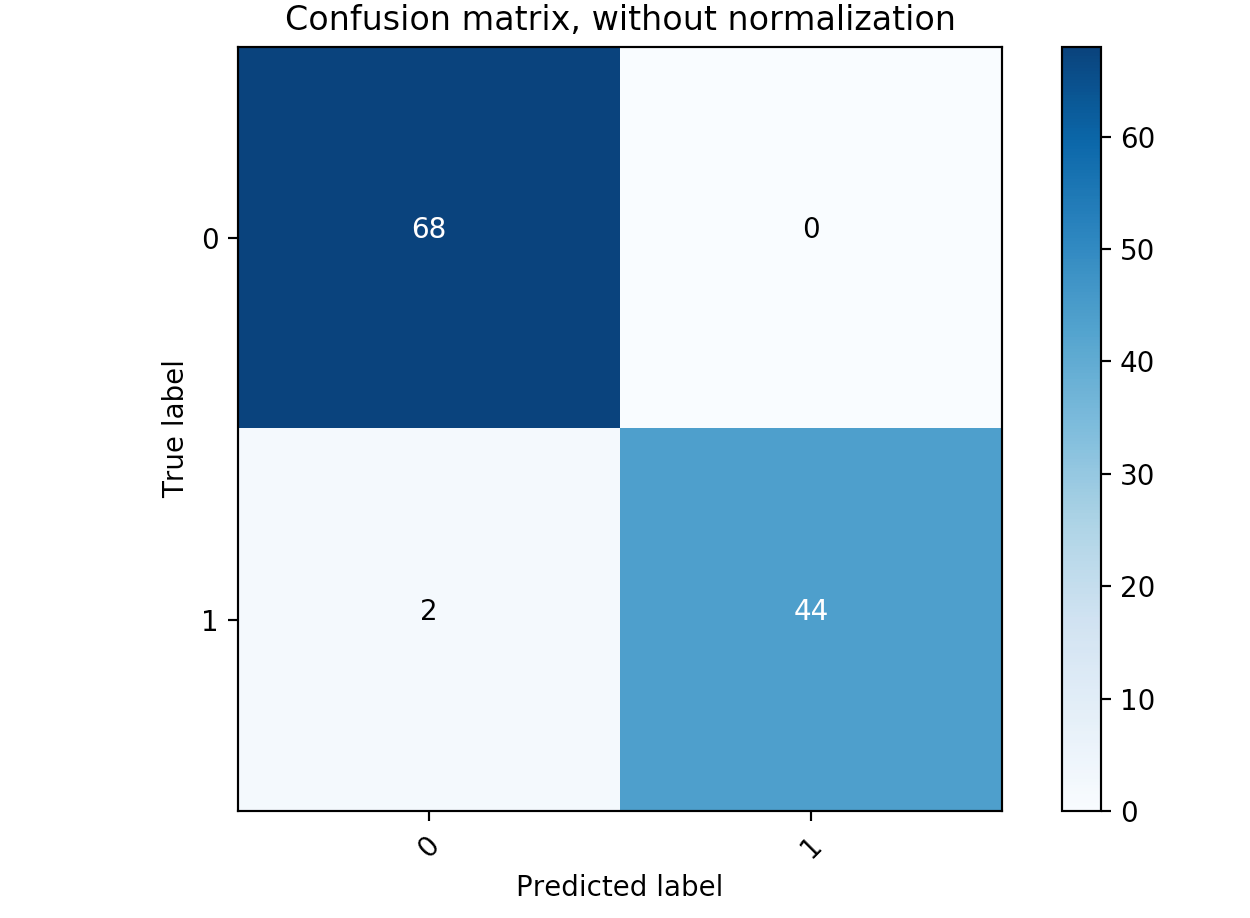


Algorithm Description:

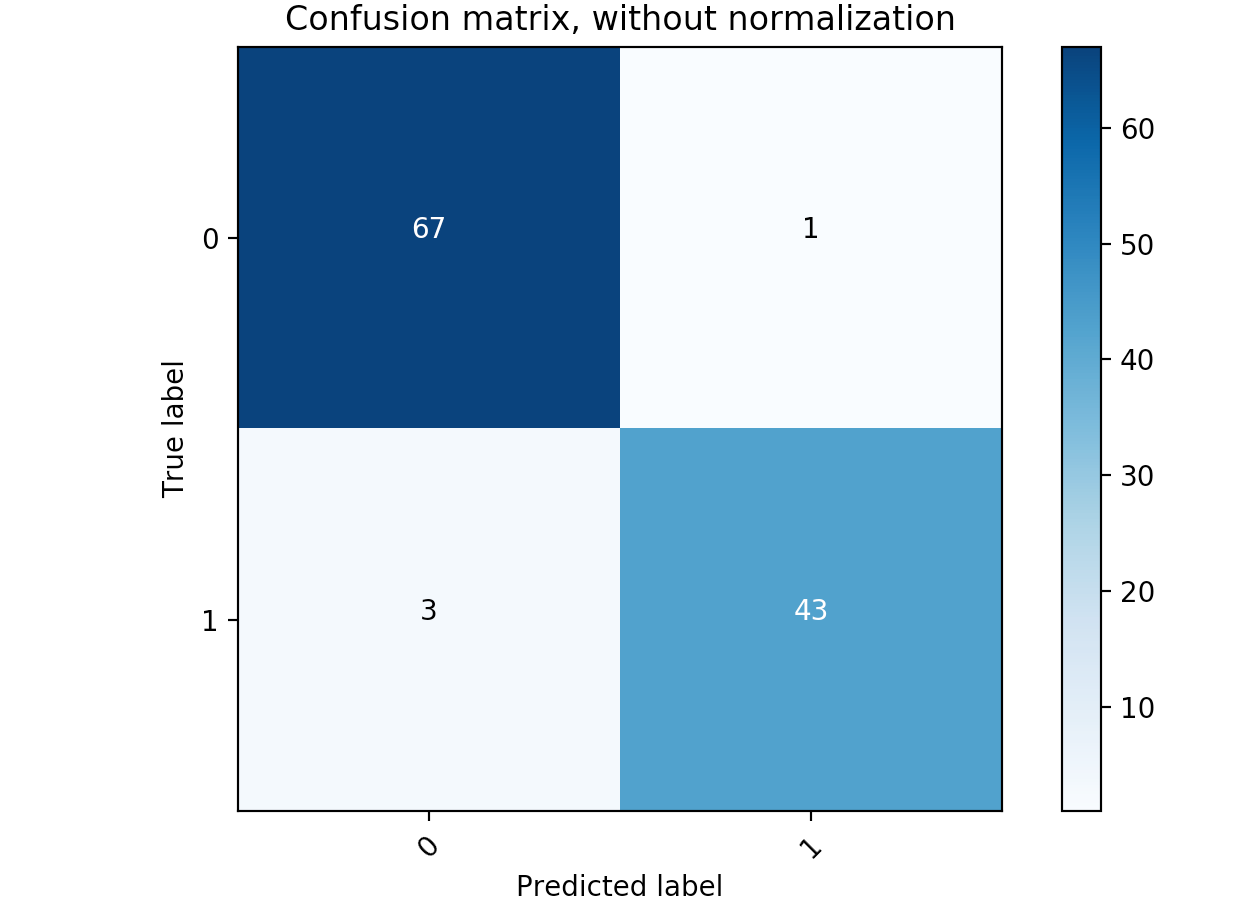
In the data preprocessing stage, the variance of each feature is really large. I standardize all the features to make the influence of them to model equal. It turns out that data without standardization or with standardization influence final prediction result a lot. Without standardization, model prediction rate is only 50%, which is random prediction. At last, I used svm as my algorithm. In svm algorithm, there are some parameter we should adjust: “C”, “gamma”, and “kernel function.” The C parameter trades off correct classification of training examples against maximization of the decision function’s margin. For larger values of C, a smaller margin will be accepted if the decision function is better at classifying all training points correctly. A lower C will encourage a larger margin, therefore a simpler decision function, at the cost of training accuracy. In other words, C behaves as a regularization parameter in the SVM. For the kernel parameter, I use “linear” and “RBM.” For gamma, it’s the parameter for “RBM.”

Algorithm Results:

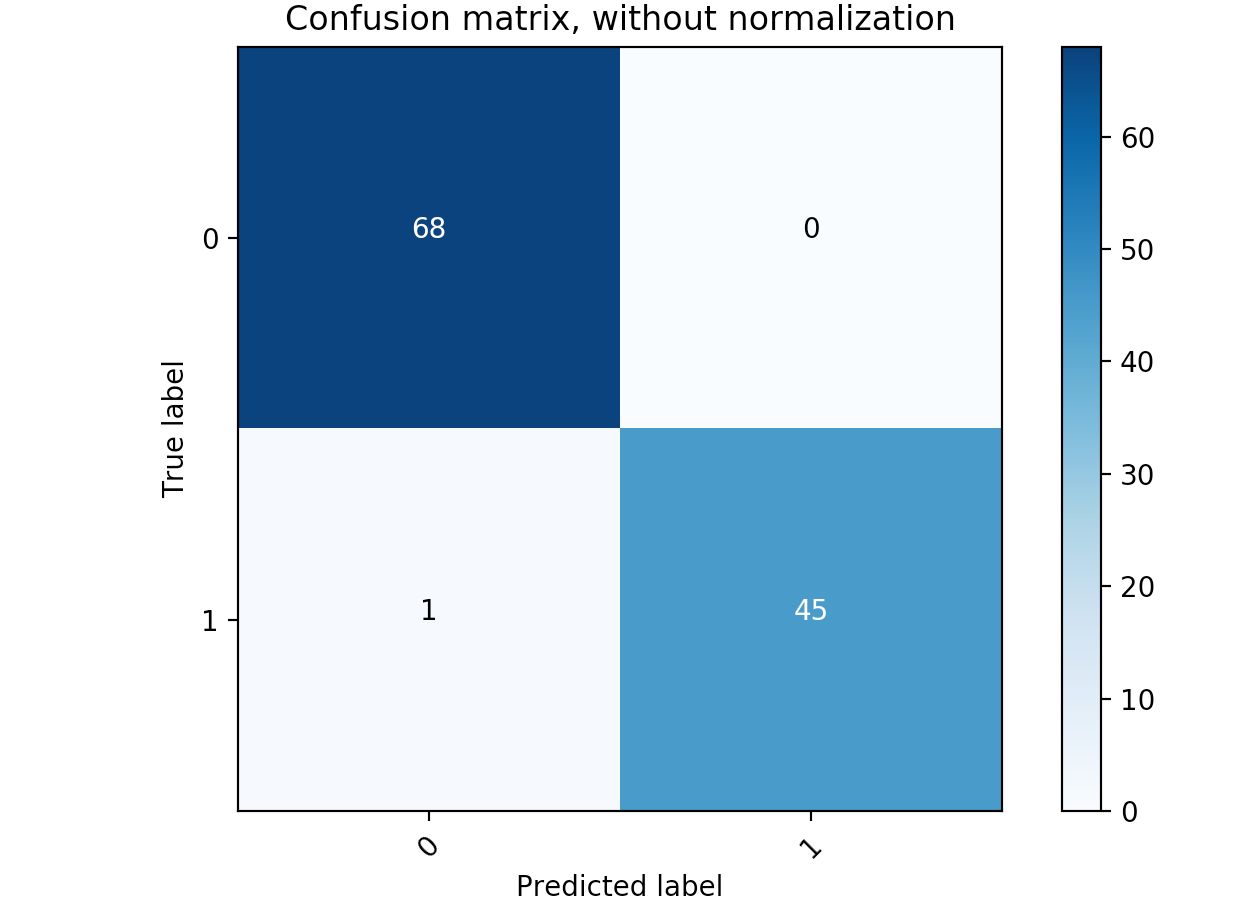
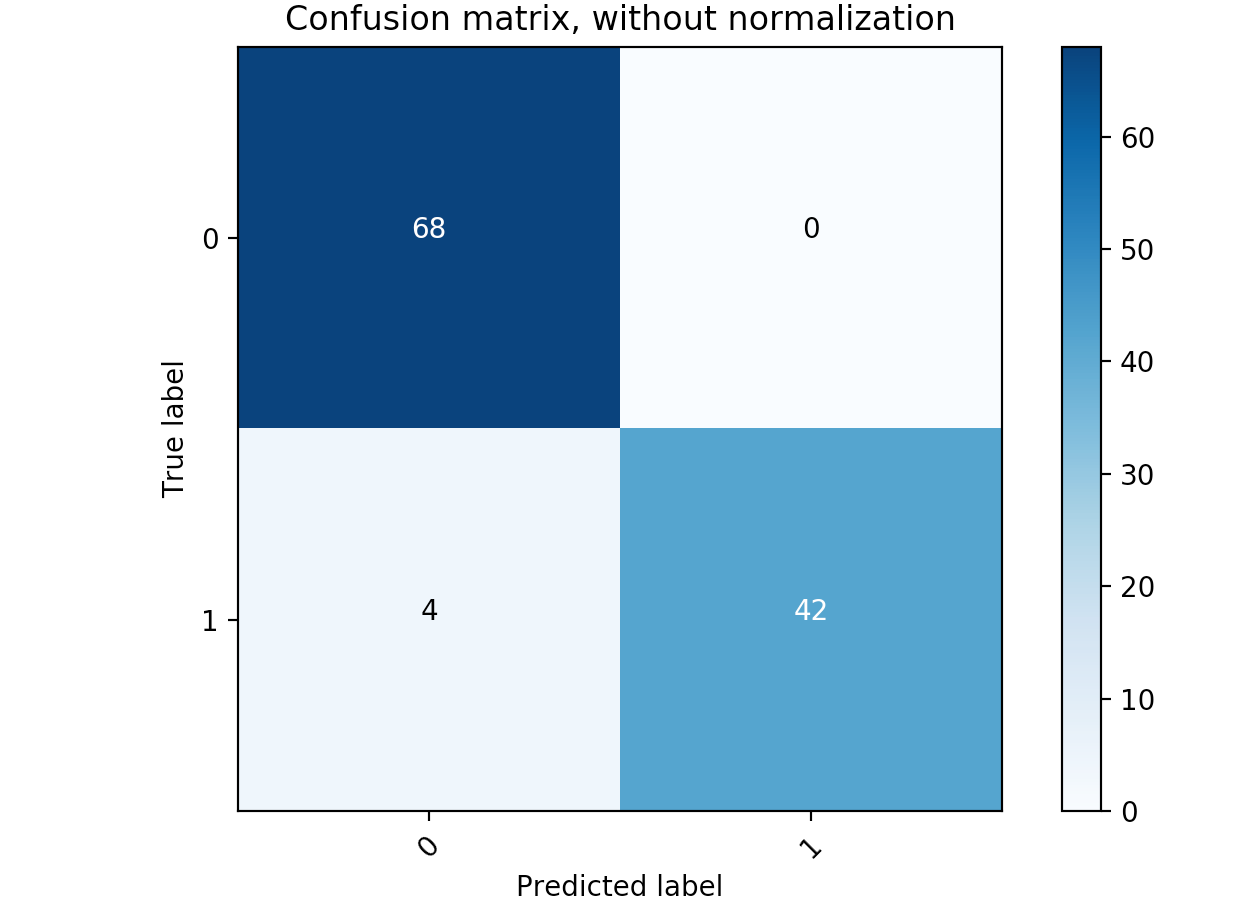
Kernel=linear, C =1, accuracy :97.6% Kernel=linear, C =0.1, accuracy :98.2%

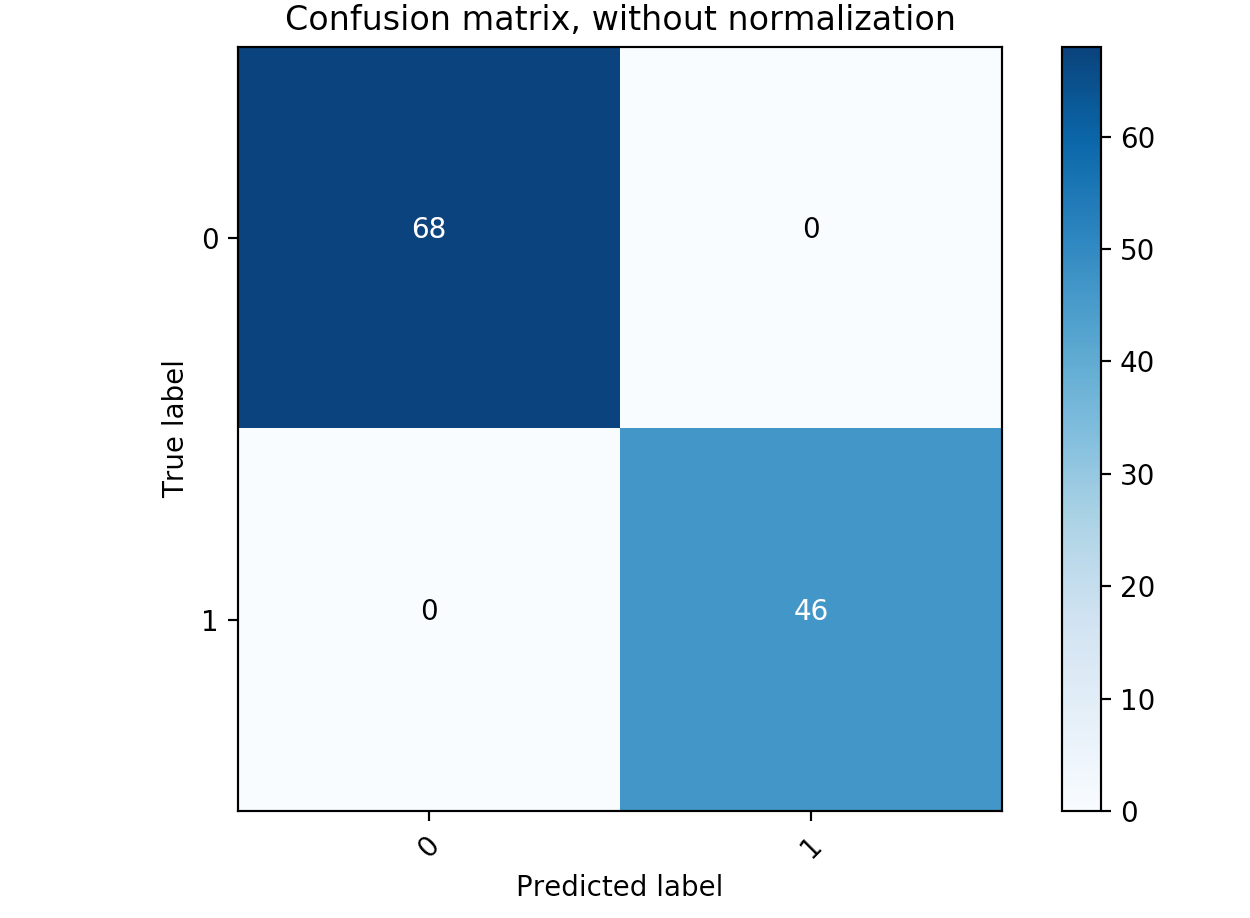
Kernel=linear, C =1, accuracy :96.5%



Kernel=RBF, C =1, accuracy :99.1% Kernel=RBF, C =0.1, accuracy :96.5%

Kernel=RBF, C =10, accuracy :100%



Runtime:

When Kernel = linear, C =1:

training\_time 0.00420s prediction\_time 0.0053s

When Kernel = linear, C =0.1:

training\_time 0.0034s prediction\_time 0.005s

When Kernel = linear, C =10:

training\_time 0.0065s prediction\_time 0.008s

When Kernel = RBF, C =1:

training\_time 0.0044s prediction\_time 0.007s

When Kernel = RBF, C =0.1:

training\_time 0.0068s prediction\_time 0.012s

When Kernel = RBF, C =10:

training\_time 0.0043s prediction\_time 0.007s

Code link:

<https://github.com/Steven0818/2019GWU_ML/tree/master/project1>