**Homework 2**

**Student ID**: 2020321249

**Name**: 남혜린 Hyelin Nam

**(a)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hw#1 dataset | Wine dataset | Letter Recognition dataset |
| Number of samples | 1000 | 178 | 20000 |
| Number of features | 2 | 13 | 16 |
| Number of classes  (Number of samples per class) | 2  (600 / 400) | 3  (59 / 71 / 48) | 26  (700-800) |
| Explanation | When plotted on a two-dimensional plane, the samples appear to form clusters by classes  C:\Users\hyeli\AppData\Local\Microsoft\Windows\INetCache\Content.Word\traindata.jpg | Each three types of wines have 13 features of chemical analysis | 16 features of each 26 capital letters in the English alphabet, composed of black-and-white rectangular pixel displays. |

**(b)**

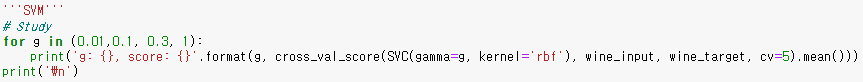
For fair cross validation scoring, test set and train sets of 5-fold trials must have samples of all classes. Given data itself were sorted in classes, so I shuffled dataset first and processed 5-fold cross validation test.

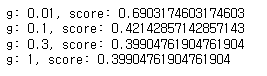
(i) SVM

In SVM algorithm, optimization process operates to maximize margin between linear decision boundary and data samples, and it uses kernel to enable linearity of deicision boundary by converting to higher dimensions. It finds optimal weights of linear boundary with all dataset with radial basis function (rbf), or Gaussian kernel.



Rbf kernel has paramter gamma, which decides distance of each data samples influences in determining the decision boundary. It is inversely proportional to standard deviation of Gaussian distribution, thus the influential distance gets smaller when gamma gets bigger, recurring under-fitting. When studied with few parameters, smaller gamma performed better. The best gamma is 1/number of features.



 result

(ii) TER-RM

TER performs overwhelmingly when numbers of samples are imbalanced between classes, because TER considers the imbalance by applying weighted summation of mean sqare loss according to the number of data in each class in the object function. When the data are balanced across classes, it operates the same with linear regression. In this experiment, is ignored. (def: TER)

RM, reduced polynomial model reduces number of weight parameters thus saving computation resource. To form RM, I calculated data correspondingly to form matrix, where is number of data samples and , =model orders, d= number of features. Then applying converted data to linear regression model forms RM model. (def: RM\_data)

I performed multiple experiments with different orders, but order=1 equals to normal SVM, so I excluded that experiment.

(iii) Linear Regression

Linear Regression finds linear line to optimally express the data sample. Therefore, the output of linear regression model does not predict the exact class. To get predicted class, I found the closest value among classes of each output. (def: preds2labels)

(iv) Comparison of methods

To make a fair comparison of the methods, first, I compared each models optimized with minimizing MSE loss, as the original method of SVM. In this experiment, RM with MSE optimization is operated, not weighted object function by distribution of classes.

I also compared TER minimizing methods. However, SVM itself is an algorithm using MSE loss, so I just scored with TER function with trained SVM model for comparison.

**(c)**

(i) Hw1 dataset

- MSE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methods | SVM | RM | | Linear Regression |
| Score  (Accuracy ) | 0.9460 | order=2 | 0.7329 | 0.7004 |
| order=3 | 0.7662 |
| order=4 | 0.7771 |
| order=5 | 0.7687 |

- TER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methods | SVM | RM | | Linear Regression |
| Score  (loss, value of object function ) | 0.1241 | order=2 | 0.1471 | 0.1559 |
| order=3 | 0.1553 |
| order=4 | 0.1417 |
| order=5 | 0.1391 |

|  |  |  |
| --- | --- | --- |
| Methods | SVM | TER-RM (order=5) |
| Confusion matrix | C:\Users\hyeli\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\52D455EC.tmp | C:\Users\hyeli\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\FCC6EEDA.tmp |

|  |  |
| --- | --- |
| Methods | Linear Regression |
| Confusion matrix | C:\Users\hyeli\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\32F36BF8.tmp |

(ii) Wine dataset

- MSE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methods | SVM | RM | | Linear Regression |
| Score  (Accuracy ) | 0.4439 | order=2 | 0.6998 | 0.8815 |
| order=3 | 0.7732 |
| order=4 | 0.5658 |
| order=5 | 0.6614 |

- TER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methods | SVM | RM | | Linear Regression |
| Score  (loss, value of object function ) | 1.8584 | order=2 | 0.4159 | 0.1639 |
| order=3 | 0.3021 |
| order=4 | 0.5946 |
| order=5 | 0.7252 |

|  |  |  |
| --- | --- | --- |
| Methods | SVM | TER-RM (order=5) |
| Confusion matrix | C:\Users\hyeli\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\3D8CCC6.tmp | C:\Users\hyeli\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\91FBCC4.tmp |

|  |  |
| --- | --- |
| Methods | Linear Regression |
| Confusion matrix | C:\Users\hyeli\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\54C37372.tmp |

(iii) Letter Recognition dataset

- MSE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methods | SVM | RM | | Linear Regression |
| Score  (Accuracy ) | 0.9723 | order=2 | 0.4181 | 0.2847 |
| order=3 | 0.4404 |
| order=4 | 0.4821 |
| order=5 | 0.4907 |

- TER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methods | SVM | RM | | Linear Regression |
| Score  (loss, value of object function ) | 0.7250 | order=2 | 24.29 | 24.75 |
| order=3 | 24.22 |
| order=4 | 24.13 |
| order=5 | 24.10 |

|  |  |  |
| --- | --- | --- |
| Methods | SVM | TER-RM (order=5) |
| Confusion matrix | C:\Users\hyeli\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\B18D9450.tmp | C:\Users\hyeli\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\B95D0EDE.tmp |

|  |  |
| --- | --- |
| Methods | Linear Regression |
| Confusion matrix | C:\Users\hyeli\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\26FFE9C.tmp |

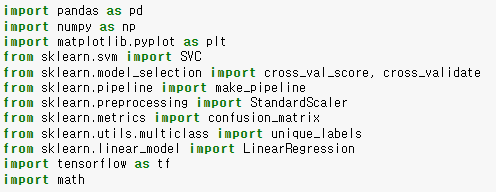
In HW1 dataset, TER-RM performs than the other methods as can be seen in overall evaluations. It overwhelms linear regression, which can be seem that RM performs effectively.

When it comes to dataset with more features, wine dataset, TER-RM results as good as SVM in prediction, but it degrades in score. I mainly used pre-made library where exact calculation scheme is not clear, but confusion matrix shows that TER-RM performs perfectly as SVM but with less computation, and even overwhelms linear regression.

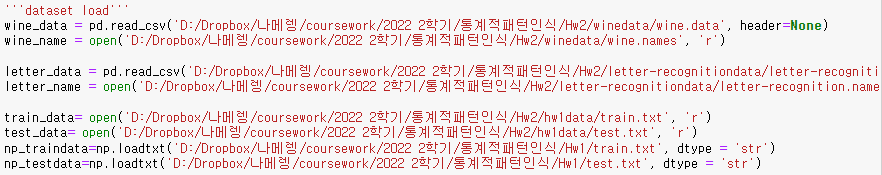
Experiment with Letter Recognition dataset, which has large feature and much more classes than wine dataset, the performance of TER-RM decreases. When the dataset is confused, it classifies more exactly when converting the data into infinitely many dimensions through the rbf kernel and then obtaining a decision boundary with a large margin, rather than trying to represent the data into a linear plane.

**(d) Appendix**

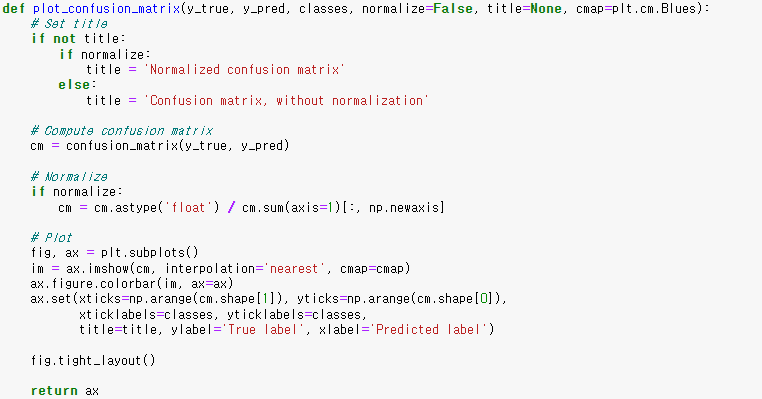
(1) Importing necessary libraries



(2) load datasets



(3) def: plot\_confusion\_matrix



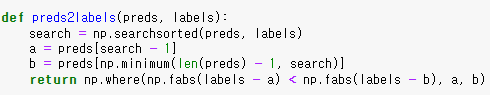
(4) def: RM\_data

Converting data to form RM



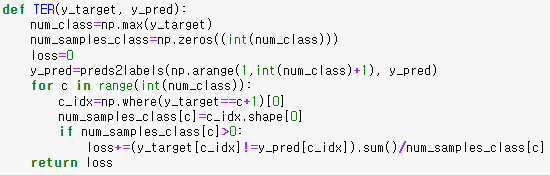
(5) def: preds2labels

Making a prediction with output of linear regression or RM model, by finding the closest label value.

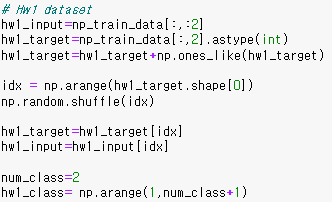
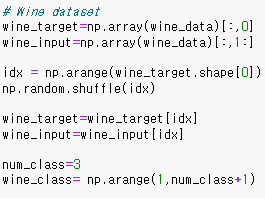
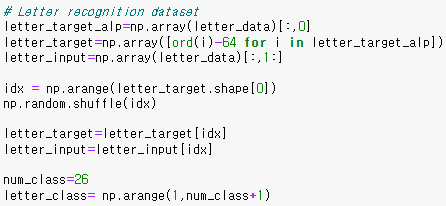


(6) def: TER

Weighted sum of MSE loss of FN and FP

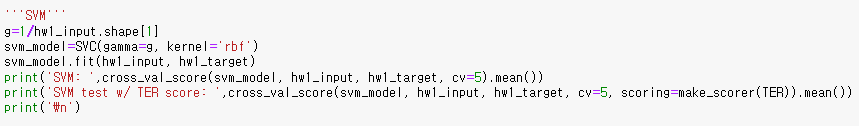


(7) Dataset pre-processing (shuffling)

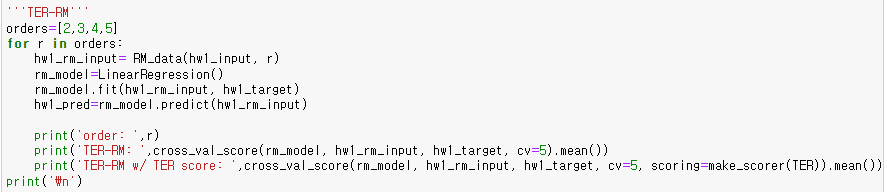
  

Converted alphabet class to corresponding int value for training.

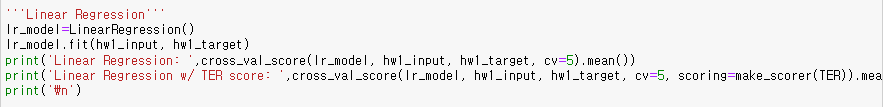
(8) SVM (with HW1 dataset)



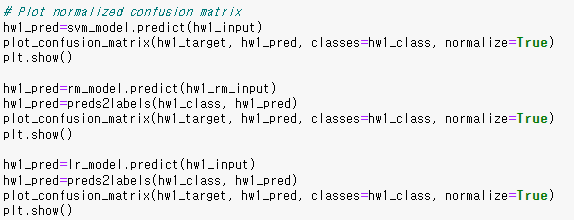
(9) TER-RM (with HW1 dataset)



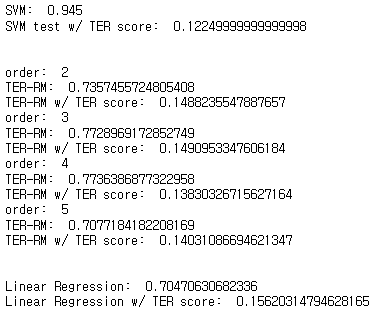
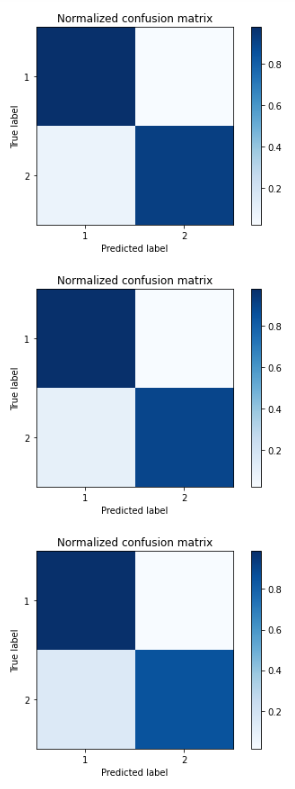
(10) Linear Regression (with HW1 dataset)



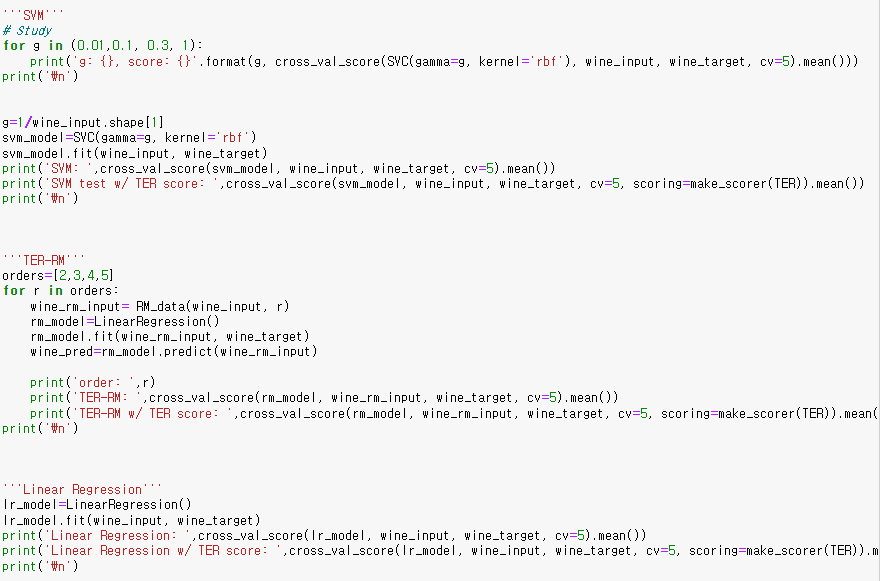
(11) Plot each confusion matrix (with HW1 dataset)



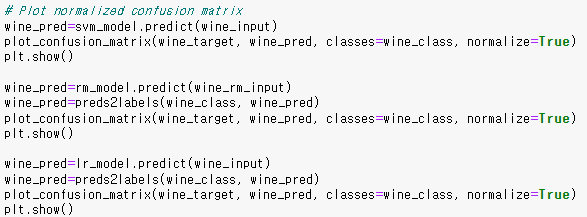
(12) Output (with HW1 dataset)

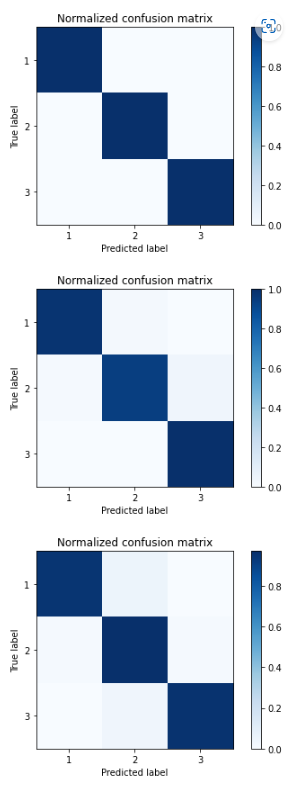
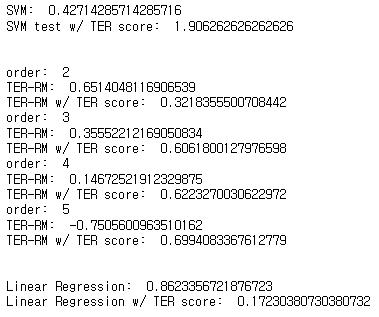
(13) Experiments (with wine dataset)



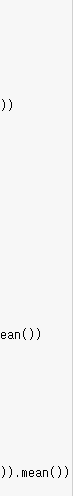
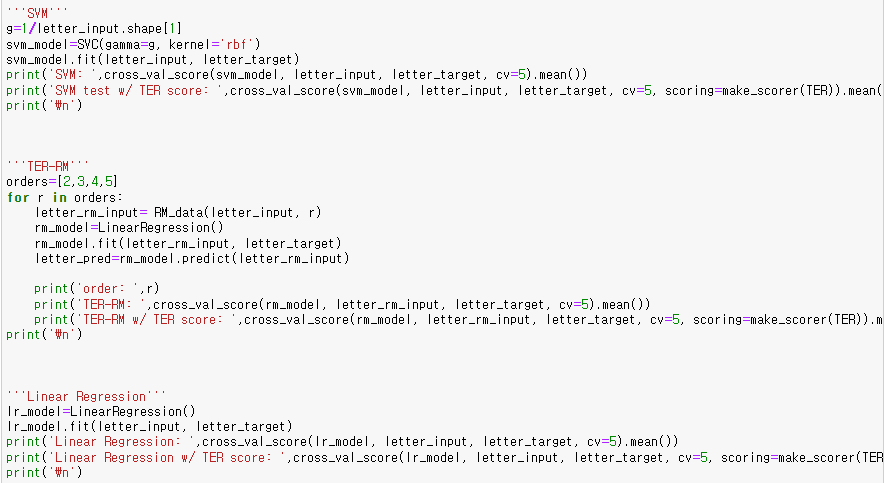
(14) Plot each confusion matrix (with wine dataset)



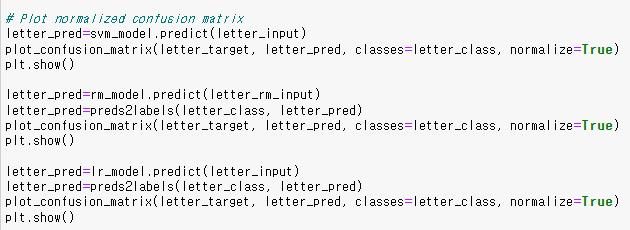
(15) Output (with wine dataset)



(16) Experiments (with letter dataset)



(17) Plot each confusion matrix (with wine dataset)



(18) Output (with wine dataset)

