**ECE 8527: Introduction to
Machine Learning and Pattern Recognition**

# HW No. 8: LDA, K-NEAREST NEIGHBORS and k-means Clustering

The short descriptions are often the longest assignments ☺

For data sets 08, 09 and 10, described in this paper:

https://isip.piconepress.com/publications/conference\_proceedings/2021/ieee\_spmb/auto\_tuning/paper\_v25.docx

and located here:

https://isip.piconepress.com/courses/temple/ece\_8527/resources/data/set\_08/

https://isip.piconepress.com/courses/temple/ece\_8527/resources/data/set\_09/

https://isip.piconepress.com/courses/temple/ece\_8527/resources/data/set\_10/

implement LDA, k-Nearest Neighbors (KNN), random forests (RNF) and k-Means Clustering (KMN). For KNN, plot performance as a function of $k$, the number of nearest neighbors. For KMN, use the same number of clusters per class, and plot performance as a function of the number of clusters per class. Select the value for each that optimizes performance on /dev, and use this to classify /train, /dev and /eval. Add the results to your table.

Include in this table results for your previous implementation of PCA (class-independent), QDA, as well as multi-class LDA, and compare your results to the results in Table 3 in the above paper. You are only allowed to train on the data labeled “train”.

Start by plotting the data and include these plots in your solution. Plot the training (/train), the development data (/dev) and evaluation data (/eval) side-by-side using the same scales for your axes (three separate plots). Next, compare the decision surfaces your implementations find on the eval data. Plot these side-by-side using the same scale for the plots (three separate plots).

Be sure to include **a single table** that compares performance, using this table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DS** | **System** | **Training Data** | **Train** | **Dev** | **Eval** |
| #08 | KNN (Paper) | /train | 23.48 | 26.62 | 64.18 |
| RNF (Paper) | /train | 29.23 | 29.45 | 59.77 |
| CI-PCA | /train | 00.00 | 00.00 | 00.00 |
| QDA | /train | 00.00 | 00.00 | 00.00 |
| LDA | /train | 00.00 | 00.00 | 00.00 |
| KNN | /train | 00.00 | 00.00 | 00.00 |
| RNF | /train | 00.00 | 00.00 | 00.00 |
| KMN | /train | 00.00 | 00.00 | 00.00 |
| CI-PCA | /train + /dev | 00.00 | 00.00 | 00.00 |
| QDA | /train + /dev | 00.00 | 00.00 | 00.00 |
| LDA | /train + /dev | 00.00 | 00.00 | 00.00 |
| KNN | /train + /dev | 00.00 | 00.00 | 00.00 |
| RNF | /train + /dev | 00.00 | 00.00 | 00.00 |
| KMN | /train + /dev | 00.00 | 00.00 | 00.00 |
| #09 | KNN (Paper) | /train | 2.11 | 3.81 | 16.63 |
| RNF (Paper) | /train | 2.06 | 3.82 | 18.32 |
| CI-PCA | /train | 00.00 | 00.00 | 00.00 |
| QDA | /train | 00.00 | 00.00 | 00.00 |
| LDA | /train | 00.00 | 00.00 | 00.00 |
| KNN | /train | 00.00 | 00.00 | 00.00 |
| RNF | /train | 00.00 | 00.00 | 00.00 |
| KMN | /train | 00.00 | 00.00 | 00.00 |
| CI-PCA | /train + /dev | 00.00 | 00.00 | 00.00 |
| QDA | /train + /dev | 00.00 | 00.00 | 00.00 |
| LDA | /train + /dev | 00.00 | 00.00 | 00.00 |
| KNN | /train + /dev | 00.00 | 00.00 | 00.00 |
| RNF | /train + /dev | 00.00 | 00.00 | 00.00 |
| KMN | /train + /dev | 00.00 | 00.00 | 00.00 |
| #10 | KNN | /train | 7.63 | 38.83 | 33.44 |
| RNF | /train | 2.15 | 39.74 | 33.28 |
| CI-PCA | /train | 00.00 | 00.00 | 00.00 |
| QDA | /train | 00.00 | 00.00 | 00.00 |
| LDA | /train | 00.00 | 00.00 | 00.00 |
| KNN | /train | 00.00 | 00.00 | 00.00 |
| RNF | /train | 00.00 | 00.00 | 00.00 |
| KMN | /train | 00.00 | 00.00 | 00.00 |
| CI-PCA | /train + /dev | 00.00 | 00.00 | 00.00 |
| QDA | /train + /dev | 00.00 | 00.00 | 00.00 |
| LDA | /train + /dev | 00.00 | 00.00 | 00.00 |
| KNN | /train + /dev | 00.00 | 00.00 | 00.00 |
| RNF | /train + /dev | 00.00 | 00.00 | 00.00 |
| KMN | /train + /dev | 00.00 | 00.00 | 00.00 |

**Make sure all entries in your table show percentages and use exactly two decimal points of precision.**

Next, pool the training data and the development data, train CI-PCA, CD-PCA and LDA on the pooled data, and evaluate on /train, /dev and /eval. Add these results to your table.

Did the results change significantly? Analyze your results and discuss why one algorithm is better than the other. Explain this in terms of how the algorithm’s assumptions about the data match the data.

Discuss the differences in performance and analyze why these differences exist for each of the data sets processed.