**Fourth Industrial Revolution (4IR) Summer School**

**Machine Learning** – **D**ay **5** exercises

**Question 1 [LDA Classification]**

Scikit learn shipped with some famous datasets such as iris, house-prices, and cancer data. In this exercise, we are going to explore Cancer dataset.

A screenshot of a cell phone

Description generated with high confidence

This dataset contains 30 different features. For the purpose of this exercise, please pick two features “**mean radius**, and **mean perimeter**” and used as a new dataset to perform the following tasks.

1. Check the dataset if it has NAN values and clean them.
2. Find out the class distribution. Is the data imbalanced?
3. Load Linear Discriminant Analysis (LDA) classifier from SciKit learn library
4. Divide the data into two sets, then fit the LDA model using one part of the separate data and test on the second one.
5. Compute and plot the confusion matrix and figure out the classifier mistakes
6. Compute accuracy, precision and recall scores of the model on test dataset.
7. Plot the decision boundary.

**Question 2 [Classification Report]**

Usually, confusion matrix gives deeper insights of the model performance in terms of shown errors. However, summarizing the performance will save us much time. In case of multi-class classification, we need to compute the summarization metrics per class. Therefore, Python provide a method called classification report that produces the summarization in a better way. The following code-snippet shows an example of using this method. Use classification report to show the model performance of the pervious exercise.

A screenshot of a cell phone

Description generated with high confidence

**Question 3 [Improve LDA Performance]**

Even though, the classification results in the previous exercise was high, but we can ask this question; can we do better? We may notice from the previous exercise that the LDA did some mistakes to classify some data points in the testing set. In this exercise, we want to improve that performance.

We want to explore two suggestions:

1. Standardize your dataset before use
2. Reduce dimensionality of the dataset

**Question 4 [Support vector machines]**

Compare the performance of SVM and LDA using the data in Question 1. Then, do model selection to identify best value of **C** parameter. (Use cross-validation to determine the best C value)

**Question 5 [Facies classification dataset]**

The dataset was prepared in a University of Kansas class exercise[[1]](#footnote-1). It consists of seven features five wire line log curves include gamma ray (GR), resistivity logging (ILD\_log10), photoelectric effect (PE), neutron-density porosity difference and average neutron-density porosity (DeltaPHI and PHIND). Note, some wells do not have PE. Also, it has two geologic constraining variables: nonmarine-marine indicator (NM\_M) and relative position (RELPOS). Moreover, it contains facies labels at half foot depth intervals. There are 9 facies labels as described below:

1. Nonmarine sandstone
2. Nonmarine coarse siltstone
3. Nonmarine fine siltstone
4. Marine siltstone and shale
5. Mudstone (limestone)
6. Wackestone (limestone)
7. Dolomite
8. Packstone-grainstone (limestone)
9. Phylloid-algal bafflestone (limestone)

A picture containing wall, indoor

Description generated with very high confidence

Given the above dataset, compare both LDA vs. SVM performance using all features of the dataset. Perform cross validation score with 5 folds to compare between them. For the wining classifier, can we improve its performance more.

You may do

1. Feature normalization.
2. Use Kernel tricks
3. Use LDA as feature reduction then apply SVM on the reduced feature space
4. Finetune the parameters.
5. Plot the cross-validation performance per fold.

**Question 6 [Semi-supervised: Label propagation]**

Scikit learn shipped with a great algorithm for various machine learning problems. It has two implementations; **label propagation**, and **label spreading**, for Semi-supervised problems. Suppose we have not completely labeled dataset. We know only 10% of the labels for sure. Then, we want the machine to help us label the reset of the dataset. Label propagation/spreading can be used in this situation.

Suppose we have the dataset shown in the figure below. The labels known to us are either in red or green. All black dots are not labeled. In this exercise test both label propagation and label spreading to handle this situation. (note: you may finetune the parameters of both algorithms to get the best labeling results)

A screenshot of a cell phone

Description generated with high confidence

**(optional section)**

**Question [LDA Classifier]**

LDA technique can be used for both dimensionality reduction and classification. It follows some clear easy steps. Given some dataset, we can follow these steps to compute the need parameters. In this exercise, we want to build LDA classifier from scratch.

1. Compute the frequency of each class
2. Compute the mean of each group
3. Compute the covariance matrix of each group
4. Compute the pooled covariance matrix
5. Compute the inverse of
6. Compute the Line Combination parameters
7. Apply the decision rule on the new data to classify

**Note: you can generate blobs dataset with two features for this exercise**

**Question [Support Vector Machines]**

Generate a 2-d blobs dataset to test Linear support vector machines classifier. Build an SVM model with **hard margin** (C parameter must be large) and plot the decision boundary as shown below. (note: the plotting code can be found in the supplements)

A close up of a logo

Description generated with very high confidence

A close up of a map

Description generated with high confidence

**Question [Linear SVM and Noise]**

Modify the data given in **Question 5** and change some labels of the data from circles to triangles and vice versa**.** Train the SVM with the same parameters (**Question 5).** Then, plot the decision boundary and observe the SVM behavior in hard margin setting. Now, relax the condition and create an SVM with a soft margin. Repeat the classification and plot the decision boundary.

**Question [Circles dataset, Kernels]**

Real data usually is not linearly separable and suffer from various issues. Generate a circles data set as shown below and build an SVM model to find the decision boundary correctly. (dataset noise = 0.2)

A close up of a piece of paper

Description generated with very high confidence

A screenshot of a cell phone

Description generated with very high confidence

1. <http://www.people.ku.edu/~gbohling/EECS833/> [↑](#footnote-ref-1)