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

**Machine Learning** – **D**ay **4** exercises

**Question 1 [Classification: Generated data]**

In this exercise, we want to study the behavior of both classifiers (kNN and Logistic regression) on some controlled datasets. Please perform the following tasks

A close up of text on a white background

Description generated with high confidence

1. Load logistic and kNN classifiers from SciKit learn library
2. Generate data for training and specify the random\_state = 20, and generate another data for testing use random\_state=2.( for blobs add this parameter: ***centers=[[1,1], [3,3]]*** ). Also, specify the size of the data for training to be 50 samples and use 10 samples for testing.
3. Perform training of the models on the training dataset
4. Compute accuracy score on test dataset
5. Plot the decision boundary of kNN
6. Plot the decision line of Logistic Regression
7. Study the behavior of the classifiers on different datasets. And compare between logistic and kNN classifiers.
8. In case of circles dataset, was logistic regression classifier successful? if not, can you provide a solution.

**Question 2 [Classification: Multi-class]**

Both logistic and kNN classifiers can deal with multi-class problems. Logistic regression original is defined for two-class classification problems. However, it can handle multi-class problems using one-versus-all style. Use blobs to generate some dataset with 5 classes. Set the parameters of the make\_blobs as given below and repeat Question 1 analysis:

* Number of samples (50)
* Number of features (2)
* Data standard deviation (2)
* Determine the random state (20)

A screenshot of a cell phone

Description generated with high confidence

**Question 3 [Classification: Curse of dimensionality]**

In some cases, we’re facing high-dimensional datasets. For example, if the problem at hand is to analyze grayscale images of size 100x100 pixels then, the problem has 10K feature dimension space. If the images are RGB-colored, the dimensionality increases 3 times. To examine, how both classifiers will work in such space, let us create two datasets, training (50 examples), and testing (10 examples). Increase the feature dimensionality to 200 features. Also, increase the spread to 7. Study the behavior of both classifiers.

**Question 4 [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)

Examine the supplemented CSV file ‘**facies\_vectors’**. Perform necessary data cleaning and prepare the data for training and testing. Finally, train both kNN and logistic regression models and deploy them to classify the testing data. Compute the accuracy of your experiments and compare it with the null accuracy.

A picture containing wall, indoor

Description generated with very high confidence

**Question 5 [Understand model mistakes]**

To understand the ML model mistakes, we need to compute the confusion matrix of the model performance. In this, exercise, refer to your models results in **Question 4** and use the predicted results to plot the confusion matrix.

**Question 6 [Understand model mistakes]**

Given the models trained in the previous **Question 4.** Compute the following evaluation metrics

1. Precision of the testing results
2. Recall of the testing results
3. F1-score

If we split the data randomly again, what will be the results?

**Question 7 [Cross Validation]**

As the data is imbalanced, we may be lucky and get the best possible testing data chunk out of the random splitting. To overcome this doubt in results, a K-Fold and Leave-One-Out cross validation methods can be used. Prepare the data in **Question 4** for 5-fold cross validation and compute the accuracy metrics. Repeat the exercise with Leave-One-Out and compare the results.

**Question 8 [Model Selection]**

Cross validation can be used to tune and find an optimal model parameter. However, as the number of parameters to tune increases, the method becomes computationally expensive. This exercise, you should use K-fold Cross Validation to find the best K of kNN machine learning model. Use **Question 4** data for this exercise.

**Question 8 [Logistic regression: simple data]**

Generate a simple dataset using blobs of size 50 datapoints, 2 features, and with cluster standard deviation = 0.5.

A close up of a mans face

Description generated with very high confidence

1. Use Scikit-learn to load Logistic regression model
2. Split the data using the code from the slides to keep part of the dataset for validation (20%)
3. Initialize a logistic regression model and training it with a training dataset.
4. Find out the labels of the remaining 20% of the dataset (validation dataset)
5. Compute the Accuracy of your model after deployment.
6. Introduce some overlapping between the two blobs in the dataset and repeat the steps from 2 to 5. Can you find better decision line.

**Question 9 [Logistic regression: hard dataset]**

Generate another dataset using circles of size 50 datapoints, 2 features, noise =0.2, and noise=0.1, factor=0.2.

A close up of a map

Description generated with high confidenceA close up of a piece of paper

Description generated with high confidence

Then, perform the following:

1. Use Scikit-learn to load Logistic regression model
2. Split the data using the code from the slides to keep part of the dataset for validation (20%)
3. Initialize a logistic regression model and training it with a training dataset.
4. Find out the labels of the remaining 20% of the dataset (validation dataset)
5. Compute the Accuracy of your model after deployment.
6. Perform feature transformation as shown in the slides for both datasets and repeat the steps from 2 to 5.

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