**Artificial Intelligence (CS 4341)**

**Assignment 3 (150 points)**

**Due: December 1st, 11:59 PM**

# Problem 1

1. Implement your own k-means algorithm from the lecture slides using Python. (**20 points**)

**“k-means-alg.py” is the file for this.**

1. Using the k-means algorithm, cluster the data from the attached file **cluster\_data.txt**. Plot X, Y coordinates for the entire dataset. Use different symbols and colors to represent your data points for different clusters.

Label X and Y axis as 'Length' and 'Width', correspondingly. Label each cluster as “Cluster 1”,

A diagram of a clustering chart

Description automatically generated with medium confidence“Cluster 2”, etc. Explain your findings. (1**0 points**)

**In the above graph we can see 3 distinct groups as well as some of the natural groupings and shapes that resulted from the data. In cluster 2 the centroid is easily understood as it is pretty much in the center of the cluster, but cluster 1 & 3 show us the impact of less common shapes of data on a centroid. We also see that the clusters are roughly the same size, this uniform distribution hints at a consistent structure or relationship among the data points in each cluster.**

# Problem 2

1. Implement your own logistic regression with regularization algorithm from the lecture slides using Python. (**20 points**)

**File is “logistic-regression-regularized.py”**

1. Using the implemented algorithm, train and test the data from the attached file **ckd\_data.zip**.

(**20 points**)

* + Use 80% of each class data to train your classifier and the remaining 20% to test it.
  + Run different values of logistic regression regularization parameter (λ). The range of λ is from -2 to 4 and the step is 0.2
  + Plot the f-measure of the algorithm’s performance on the **training and test sets** as a

function of λ:

*f -measure*= 2*Pre**Rec Pre*+ *Rec*

## TP TP

where *Pre*= ; *Rec*= ; *TP* + *FP TP* + *FN*

A graph of a line

Description automatically generated with medium confidenceand *TP* is the number of true positives (class 1 members predicted as class 1), *TN* is the number of true negatives (class 2 members predicted as class 2), *FP* is the number of false positives (class 2 members predicted as class 1), and *FN* is the number of false negatives (class 1 members predicted as class 2).

1. Repeat the procedure in (b) but now using the features normalized with the standardization protocol discussed in the class. (**10 points**)

**File is “logistic-regression-standardized.py”**

**F-measure on Training Set (Unstandardized): 0.9975**

**F-measure on Test Set (Unstandardized): 0.9600 F-measure on Training Set (Standardized): 1.0000 F-measure on Test Set (Standardized): 0.9796**A colorful rectangular bars with black text

Description automatically generated with medium confidence

# Problem 3

Apply three clustering techniques to the handwritten digits dataset. Assume that k = 10. **(30 points)**

1. K-means clustering (implemented in Problem 1).
2. Agglomerative clustering with Ward linkage [(sklearn.cluster.AgglomerativeClustering)](http://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering).
3. Affinity Propagation [(sklearn.cluster.AffinityPropagation)](http://scikit-learn.org/stable/modules/clustering.html#affinity-propagation).

The dataset you will be working with is the handwritten digits and the details can be found [here.](http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html)

***Assess all three clustering algorithms using the following protocol:***

1. Each cluster should be defined by the digit that represents *the majority* of the current cluster. For examples, if in the second cluster, there are 60 data points of digit “5”, 40 of “3” and 25 of “2”, the cluster is labeled as “5”.
2. Report the 10x10 confusion matrix by comparing the predicted clusters with the actual labels of the datasets. If the clustering procedure resulted in less than 10 clusters, output “-1” in the position to the missing clusters in the confusion matrix.
3. A screenshot of a computer program

   Description automatically generatedA screenshot of a computer program

   Description automatically generatedCalculate the accuracy of each clustering method using the [Fowlkes-Mallows index](https://en.wikipedia.org/wiki/Fowlkes%E2%80%93Mallows_index) [(sklearn.metrics.fowlkes\_mallows\_score)](http://scikit-learn.org/stable/modules/clustering.html#fowlkes-mallows-scores).

**File is “k-means\_agglormeritave\_Affinity-prop.py”**

# Problem 4

Apply three classification algorithms to the same **ckd\_data.zip** dataset as in Problem 2 **(40 points)**

1. Support Vector Machine with the linear kernel and default parameters [(sklearn.svm.SVC)](http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html).
2. Support Vector Machine with the RBF kernel and default parameters.
3. Random forest with default parameters [(sklearn.ensemble.RandomForestClassifier)](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html).

***Assess all three classification algorithms using the following protocol:***

i. Use 80% of each class data to train your classifier and the remaining 20% to test it. ii. Report the f-measure of the algorithm’s performance on the **training and test sets.**

*f -measure*= 2*Pre**Rec Pre*+ *Rec*

## TP TP TP + FP TP + FN

where *Pre*= ; *Rec*= ;

**File is “SVMs-and-randomForest.py”**

**F-measure on Training Set (Linear SVM): 0.9801**

**F-measure on Test Set (Linear SVM): 0.9796**

**F-measure on Training Set (RBF SVM): 0.7692**

**F-measure on Test Set (RBF SVM): 0.7692**

**F-measure on Training Set (Random Forest): 1.0000**

**F-measure on Test Set (Random Forest): 1.0000**