**CS 4824 / ECE 4424 HW 4 Programming Portion**

**General Instructions:**

* For this programming assignment, we have provided a starter code. Your primary task will be to complete the code from scratch to implement K-Means Clustering. You cannot use implementations of Sklearn or other libraries for this assignment.
* We are providing you with one iPython notebook: *Clustering.ipynb*. It will use your implementation of KMeans Clustering imported from *cluster\_kmeans.py*, and will be where we look when grading. It also has the popular ML library sklearn’s KMeans implementation commented out. For your reference, feel free to uncomment and compare your solution with it.
* You need to upload *cluster\_kmeans.py*, that you will be modifying for this assignment. You also need to upload a Clustering.ipynb file for this assignment. We will be testing your code while keeping all other files the same as provided in the original folder. Please make sure that your code works with our provided iPython notebook *Clustering.ipynb*. At the top of your Python submission file and ReadMe file, please include your name and VT email id as a comment, so that it is easy for us to manage your submissions during the grading process.
* We are not providing any test cases file for this assignment. We will test your code with your submitted iPython notebook *Clustering.ipynb*.
* Before starting all the tasks, examine the entire codebase. Follow the code from both of the iPython notebooks to see which methods they call. Make sure you understand what all of the code does.

**Programming Tasks [15 Points]:**

**Task 1:** (10 points) Implement the KMeans Clustering from scratch using Class *CustomKMeans* in *custom\_kmeans.py.* Use Euclidean distance as a similarity metric in this task. Implement the *fit* method and run it using *Clustering.ipynb*. You will also need to maintain these attributes: “*inertia\_*” and “*labels\_”* to run *Clustering.ipynb*. Inertia is the sum of sample distances from their centroid. Labels represent the cluster number for each sample. Make use of plot\_state method in each iteration of KMeans. Centroids movement should be visible in the plots of every iteration in your submission of *Clustering.ipynb*.

Hint: You can make two helper functions. One is to assign data points to centroids and other is to update the centroids.

**Task 2:** (2.5 Points) Choose the best K for above task and defend your choice in the bottom of *Clustering.ipynb*. K represents the number of clusters. Hint: Elbow Method.

**Task 3:** (2.5 Points) Interpret the clusters to tell, what do each of the clusters represent? Mention that also in the *Clustering.ipynb* file. Make use of facetgrid and scatter plots.

**Note:** You can also use *plot\_3d.py* to check out the labels along the first three principal components by running `*python plot\_3d.py*` in the command line.

This assignment uses the dataset from the kaggle.com “Credit Card Dataset for Clustering” challenge. Find it, and exploratory data science notebook submissions, at https://www.kaggle.com/arjunbhasin2013/ccdata