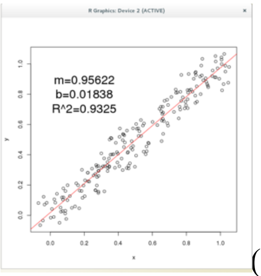
# Unsupervised Learning

Author: Timothy Morren

Project Description: The goal of the project is to create a python program that utilizes K-means clustering on labeled classification data to predict classifications of future data.

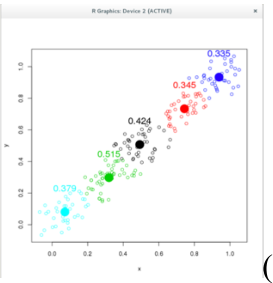
Unsupervised Learning: Unsupervised learning approaches rely on unlabeled data for training (no classes). For this project we will be using the same data as project 3 (labeled data) but we will hide the class attribute before grouping the data.  In general, unsupervised learning rely on statistical properties of the training data to develop useful or meaningful representations of concepts. Some Unsupervised learning examples are Clustering, Manifold Learning, Regression (Parametric and/or Non-parametric). Below we will compare Regression models and k-means clustering models.

Regression: Regression is one of the simplest models for prediction. Say, the data is linearly related; we can model the data with a linear regression line, or line of best fit, so that we can use that line to extrapolate the data out past known data points. For example, let's say we were given data on plant growth (up until they reach 1 centimeter tall) and the amount of sunlight each plant received. We could plot the data points and see a linear trend (Figure 1). That is, plant growth is linearly related to the amount of sunlight. With our domain, or plant growth, limited to until it reaches 1 centimeter; we must use a linear regression line to predict the growth after 1 centimeter. Thus, extrapolating the data to further unknow points.



(Figure 1.)

Clustering: What if the data is not this nice and straight but more non-linear? We must use a different approach in grouping the data in useful ways. Assume that the data can be represented using a smaller number of data points which express the average or mean behavior of the data at each point (Figure 2). This way we don't need all the data, but just theses averages that represent a group of closely related data. It also makes no assumptions about how the data will be structured.

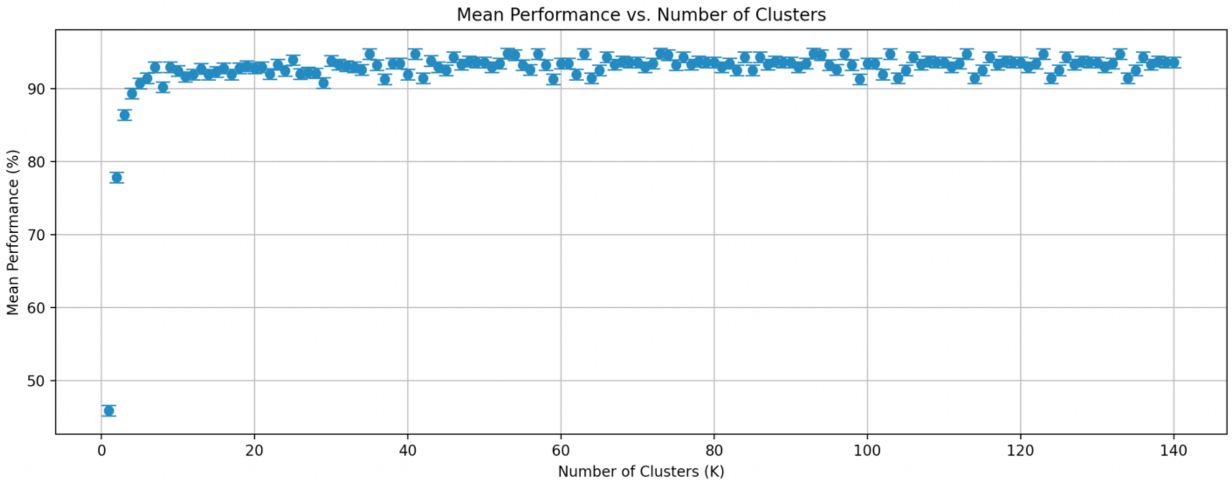


(Figure 2.)

K-means Clustering: Given a set of N training examples each consisting of a vector of continuous attributes, select K training examples to be a set of initial cluster means (centers). Step 2: Determine the distance between each training example and each of the K cluster means. Assign each training example to the closest cluster mean. Calculate the average of the training examples assigned to each cluster mean (creates a new mean). Go back to step 2 until the cluster means do not change (i.e. all training examples are assigned to the same cluster mean as on the previous iteration).

Special Cases of K: If K=1, you will just simply get the average of the data, but this might be perfect if your data is only about a single average. On the other hand, if K=N, we have too many basis vectors (centers) and haven't simplified our model description at all since there are still the same number of points. So, we want a K that will correctly classify most examples while also minimizing the number of averages needed to store, check, and update.

Testing: By running repeated random subsampling cross-validation on the code we were able to get some important benchmark statistics from the model. We calculated the mean performance of each K-value model. Mean performance, in this case, is the percentage number of correctly classified data points out of the amount of validation points it was given. This was done 100 times with each K-value and the percentages were averaged. Thus, below we have a graph of K (number of centers) vs Mean performance of using K number of centers.



(figure 3. Using Iris data)

Interpreting results: As we see from the graph above, the larger the K-value the better average the model has at predicting. But after about k=10 the average stays about constant. This means with only around 10 data points we can represent the information held within the original 150 points. A compression of the data 15 times smaller than the originally stored data.