1. The k-means clustering algorithm aims to partition n data points into k clusters (groupings) such that the distance between each point and its assigned cluster's mean is minimized. More specifically, given a data set $(x_1, x_2, x_3, ..., x_n)$ a k-means algorithm aims to assign a cluster to each data point x satisfying the objective function:

$$arg_S min \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu||^2$$

Where k is the number of clusters, S_i is the set containing data points in cluster i, and μ_i is the mean of set Si. Traditionally the L2 norm, $||x - \mu||^2$, is used; and the mean μ_i is calculated by taking the vectorial sum of the points and dividing by the cardinality of the set.

Implement a k-means algorithm from scratch (whichever one you like) and run on the provided data set for k=3 clusters. Now, change the objective function to use the L1 norm (taxicab norm) instead. You should compute the mean just the same. Your new objective function is:

$$arg_S min \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu||^1$$

Your final code should allow the user to select the norm used, to specify k, and should work on data of an arbitrary data dimension d of no greater than 100. Avoid using an ML library; instead, hand code algorithms such as L2 and L1 norm. You are free to use linear algebra libraries, such as numpy.

Finally, treat your algorithm as a model. Separate your data into a train and test set. With the train set, perform the k-means optimization and freeze the centers. Then, predict on your test set. Note the error between the predictions and the class labels for your test set.

Throughout, you can either assume knowledge of class number, or implement your own method of ascertaining a class number. (A hyperparameter search, perhaps?)

Deliverables:

1. Final code.
2. Run the code on the *iris_train.csv* dataset for both norms, and only use the first four categories. (Ignore the last category of species type.)
3. Submit a csv file that is the source dataset with an appended column of classification.
4. Submit images of the points plotted in multivariate space and color coded based on classification. (for both norms.)

5. Now import both *iris_train.csv and iris_test.csv*, and produce a model on the train dataset for each norm above. Finally give metrics of prediction accuracy on the test data for each norm.