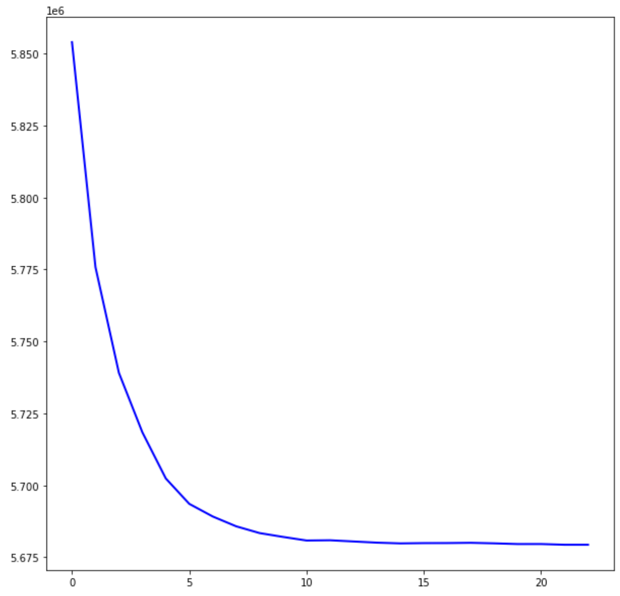
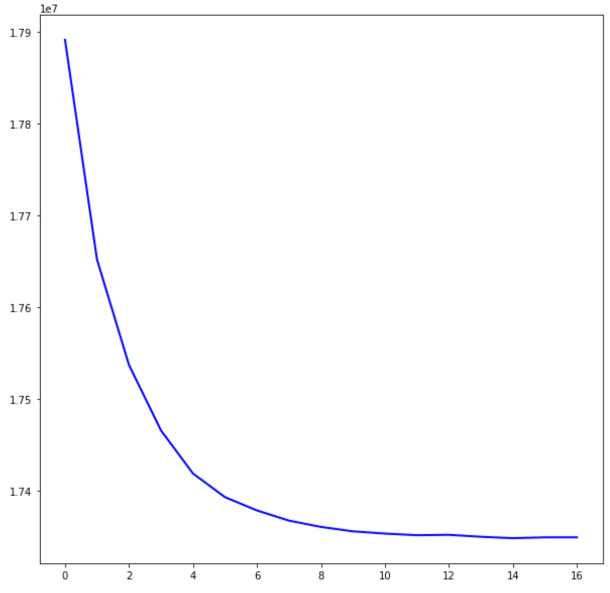
Q2. At first, randomly 3 points are chosen in the sample are chosen from within the sample which will be the initial centroids of the 3 clusters (k=3). The Euclidean distance between each sample and each of these 3 centres are calculated. The centroid with which the Euclidean distance is minimum, the datapoint belongs to that cluster. This clusters are assigned as 0, 8 or 9. Now for all the 3 clusters we find the centroid from its data points and again compare the Euclidean distances of each data points and again assign them to the particular cluster with minimum distance. This goes on iteratively until the previous cluster assignment of the datapoints are exactly same as the new cluster assignment.

The K-means algorithm took **23 iterations** to converge. Below is the graph of the error term with the number of iterations -

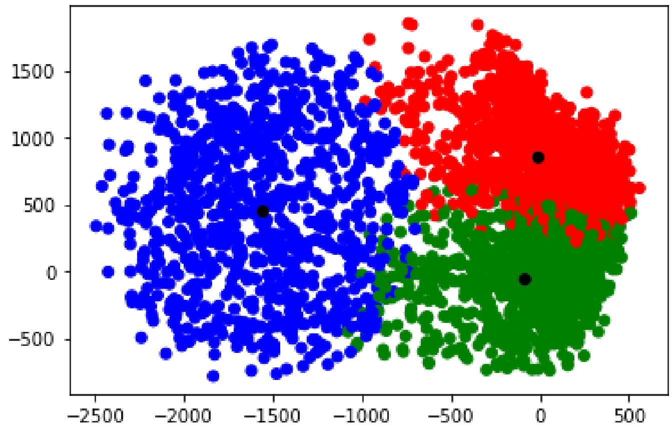


This above plot is expected as the number of iterations increase the reconstruction error decreases and after a certain number of iterations the error does not decrease anymore which results in convergence.

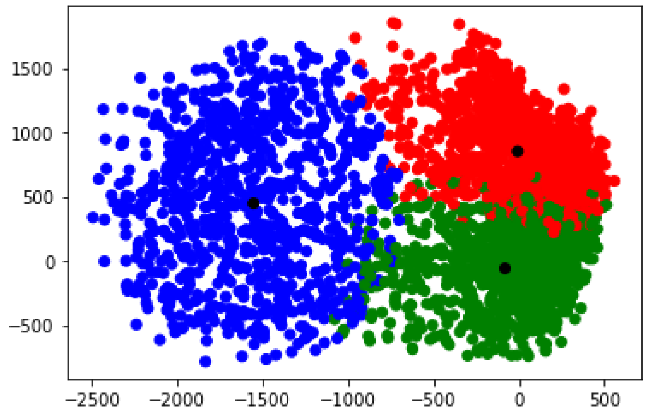
Project data into 72 dimensions with PCA converged in **17 iterations –**



K-means Clusters using PCA



K-means Clusters



Q3. Discriminant function –

$$\delta_c(x) = -\frac{1}{2}(x-\mu_c)^\top \Sigma_c^{-1}(x-\mu_c) - \frac{1}{2}\log|\Sigma_c| + \log \pi_c$$

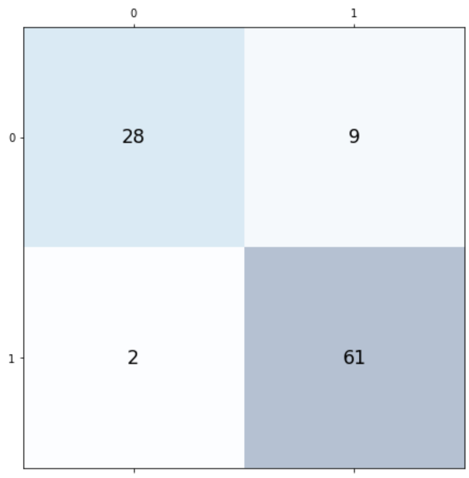
$$\mu_k = \frac{\sum_{i=1}^{n} P(z=k|x^{(i)})x^{(i)}}{n_k}$$

$$\Sigma_k = \frac{\sum_{i=1}^{n} P(z=k|x^{(i)})(x^{(i)}-\mu_k)(x^{(i)}-\mu_k)^\top}{n_k}$$

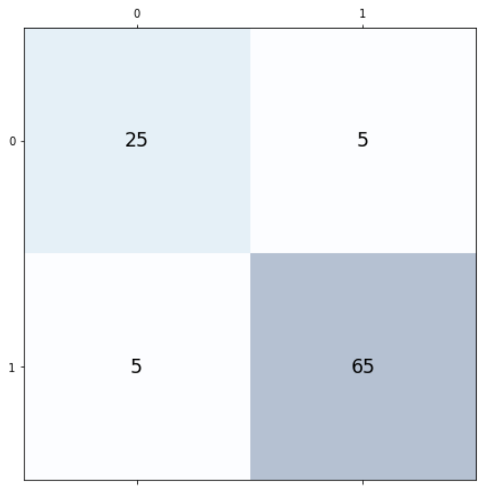
$$n_k = \sum_{i=1}^{n} P(z=k|x^{(i)})$$

Discriminant function is calculated considering it belongs to class 1 or class 2 and comparing the values which is greater the sample is concluded to belong to that cluster. Unlike K means the gaussian model conders the mean and the variance of the sample while clustering and those are calculated on the above-mentioned terms while calculating the discriminant function.

Confusion Matrix for Gaussian Discriminant with class-dependent covariance -



Confusion Matrix for Gaussian Discriminant with class-independent covariance -



The class-independent covariance and the class dependent covariance are showing different results because we are using different discriminant function for each one of them.

Q4. The dataset that was provided had 2000 samples. We utilized 1600 samples to estimate parameters and the rest 400 samples for testing. The whole dataset is vectorised into 1000 maximum features and presence of each such feature is projected into the X_train and X_test arrays and their corresponding values 0 (if the mail is not spam) and 1 (if it is spam) is stored in the y_train and y_test arrays.

The fit() function in the class NaïveBayes computes the parameters like the prior and the likelihood of each sample given a class (0 or 1). The predict() function is used to predict the class of the X_test dataset computing the logarithmic probabilities using the Bayes Rule. This returns a class value (0 or 1) for each sample. The get_prior_prob() is used to compute the prior probability of a class k, thus it returns a scalar. And lastly, the get_class_liklihood is used to calculate the Bernoulli parameter theta for each feature for each class which is basically the mean of the features in a particular class k.

This returns a vector of size d (no of features). Thus, basically we are computing the probability of each sample to be classified as a spam or not a spam given the probability of its features being in a spam or not being in a spam email.

The associated confusion matrix –