Comparing K-means and GMM Clustering Algorithms to Identify Benign and Malignant Cases in Breast Cancer Data

Introduction -

Our group decided to focus our analysis on a dataset called "Breast Cancer Wisconsin (Original)". This dataset consists of data obtained from the patients of Dr. William H. Wolberg at the University of Wisconsin Hospitals. The instances of data are split into groups that were observed over the span of two years (from 1989-1991). We chose this dataset because it contained unsupervised data which made it perfect for the learning algorithms we wanted to implement and also because the implications of our analysis would have very real applications to the medicinal world. Our analysis consists of conducting the K-means and Gaussian Mixture Model (GMM) algorithms on this dataset to predict which attributes observed are most indicative of a benign vs a malignant tumor. The attributes we considered were the following: Clump Thickness (1-10), Uniformity of Cell Size (1-10), Uniformity of Cell Shape (1-10), Marginal Adhesion (1-10), Single Epithelial Cell Size (1-10), Bare Nuclei (1-10), Bland Chromatin (1-10), Normal Nucleoli (1-10), and Mitosis (1-10). We also aim to compare the two algorithms to determine which one is a better fit for this data.For both algorithms we chose to only assign two cluster centers since our analysis is limited to classifying whether a tumor is benign or malignant. Classifying whether a tumor is benign or malignant just by observing its physical properties can be very applicable to the field of breast cancer research. For example, from our analysis if a certain attribute is more indicative of the tumor being malignant, it could serve as an immediate admonitory signal which could help doctors prioritize care as needed.

Related Works -

Several research studies have been conducted to attempt to classify Breast Cancer Data using both the Wisconsin dataset and other related datasets. We have the work of Mohammed et al. who aimed to apply and compare the performance of each model of three different Machine Learning Classification Models - Decision Tree (DT), Naive Bayes (NB) and Sequential Minimal Optimization (SMO). Taking into consideration the resampling performed to mitigate imbalanced classes in the WBC dataset, the SMO model seems to perform better than the others ("Analysis of Breast Cancer Detection Using Different Machine Learning Techniques"). The Dubey et al. study also looks into the WBC dataset, which utilises only the K-means algorithm for their analysis. They make use of both foggy and random initialization of cluster centroids. A final average positive prediction accuracy of 92% was achieved ("Analysis of K-means Clustering Approach on the Breast Cancer Wisconsin Dataset").

Methods -

As a preliminary step to our analysis, we chose to conduct a Principal Component Analysis (PCA) on our dataset to lower its dimensionality and to find the subspace in which the data has the most variance. In the original data, the attributes had values from 1-10 which not only skewed our visualizations but also made it harder to analyze our data. Moreover, because it's impossible to plot nine-dimensional data, we decided to implement PCA to visualize the clustering of our nine-dimensional and later to run the clustering algorithms on and compare to the results of running GMM and K-means on our original nine-dimensional data. To implement

PCA, we used the 'transpose trick' and created the matrix A^TA (where A is the mean subtracted data). We used numpy.linalg.eig to extract the eigenvectors and values. We then used the eigsort method to sort eigenvectors from largest to smallest corresponding eigenvalues and normalized the dot product of mean subtracted data and the sorted eigenvectors. The transpose of this result dotted with the mean subtracted data provided us with our principal component matrix. Using PCA steps, we stored the top three principal components, and added them to our dataframes in new X, Y, and Z columns so we could plot each tumor/observation in two and three dimensions.

After completing PCA, we implemented K-means clustering which we hardcoded using a similar format to the K-means method from homework two but rewritten to handle and plot multidimensional data. We wrote two K-means clustering methods both of which implemented the K-means Expectation-Maximization (EM) algorithm of randomly setting mean centers with numpy.random.permutation, calculating distances between each point and the centers to assign clusters based on which distance was shortest (using numpy.argmin to set the smallest distance equal to one and the rest to zero producing the responsibility matrix Rnk), recalculating the mean centers based on new cluster assignments and repeating this process until the mean centers converge/stop moving. Through every iteration, the data were plotted with their current cluster assignments. The difference in the two methods we wrote was that one plotted convergence in two dimensions and the other plotted in three. The two dimensional method was run using the original nine-feature data by calculating the distances between each datapoint and the mean centers in nine-dimensions and assigning clusters to each point and then using the PCA columns to plot these points. The cluster assignments derived from running K-means on the original data were stored in a new column in the dataframe called "K-means Raw". Next, the three dimensional K-means method was run on the original data as well for plotting purposes. It's important to note that both of these results would produce the same cluster assignments; the only thing that differed was how the clusters were plotted (2D vs 3D). Finally, the three dimensional K-means method was run on just the PCA data, meaning the X, Y, and Z; clusters were plotted and the results of this run were stored in a new column in the dataframe called "K-means PCA".

Our next step was implementing the Gaussian Mixture Models (GMM) algorithm on the data. GMM assumes that all data are produced from a mixture of Gaussian distributions with unknown parameters and uses covariance and centers of latent gaussians to produce confidence ellipsoids around data points. The GMM EM algorithm differs slightly from the K-means EM and is as follows: the E step is to compute the responsibilities for each Gaussian for each datapoint, and the M step is to update the parameters of each Gaussian to maximize the likelihood for the responsibilities calculated in the E step. The parameters that GMM considers are the prior probability (π_k), and the mean (μ_k) and covariance (Σ_k) of each Gaussian density. Unlike K-means, GMM calculates a "soft responsibility" for each point meaning that instead of its assignment being one or zero for each cluster, each datapoint has a probability assigned for each cluster. This difference and the fact that GMM maximises parameters instead of minimizing distance between points and clusters (like in K-means) is why our group was drawn to the comparison of the two algorithms. Our group imported the scikit-learn library to use the inbuilt mixture.GaussianMixture method. We first ran this method on the original data (with all 9 dimensions) and then we also ran it on the PCA data where the original data had been reduced to

three components. Since the GMM algorithm considers covariance as a parameter, the inbuilt GMM method also requires the user to choose between four covariance types which are: spherical, diagonal, tied, and full. To test which type was best for our data, our group plotted the GMM data using the inbuilt method using all four covariance types individually to determine which covariance reflected our dataset's shape the best. After observing the visualizations, we decided to conduct our formal analysis with the spherical covariance parameter and two as the number of components parameter (benign or malignant).

Results -

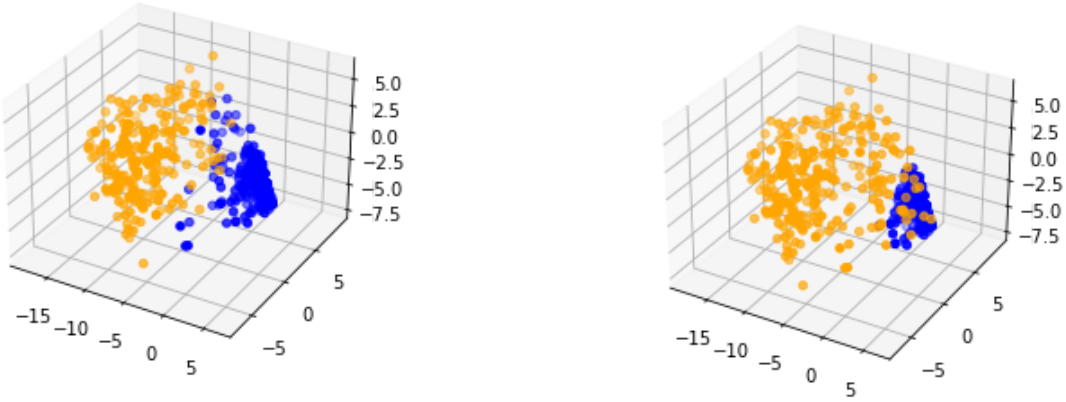
Table 1. Accuracy Table

	Kmeans Raw	**Kmeans PCA (3D)**	**GMM Raw**	**GMM PCA (3D)**
True Benign (TN)	0.9759	0.9759	0.9105	0.9410
True Malignant (TP)	0.9211	0.9211	0.9959	0.9959

We achieved an accuracy of around 95.71% when we classified our data using the K-means algorithm (both the raw data as well as the PCA-modified data had the exact same results). However, this doesn't represent the True Positive and True Negative results of the algorithm. We notice that our K-Means model predicts Benign observations with an accuracy of 97.59% as opposed to a True Malignant accuracy of 92.11%. This could possibly be associated with the general functionality of the K-means algorithm. We know that distance is the key factor in determining cluster associations for this method and since the Benign cluster appears to be more compact with the points located much closer to one another, it becomes easier for the algorithm to classify these points. On the other hand, the Malignant observations, as can be seen from the plots, appear to be more spread out and more elongated. Subsequently, the algorithm is likely to perform relatively worse. Additionally, the study conducted by Dubey et al. (referenced in Related Works), also resulted in an average positive classification accuracy of around 92% (compared to our 92.11%). Thus, the overall performance of our model is quite satisfactory. We considered evaluating our model on the basis of speed and number of iterations. However, after several runs of the K-means algorithm, we still were not able to achieve any kind of consistency in the number of iterations. This can be attributed to the random initialization of cluster centres. The initial placement of cluster centres plays a huge role in how quickly the algorithm converges to a solution.

Figure 2. Final clusters of data points obtained from both models ran on raw 9-dimensional data

K-means Raw Data Final **GMM Raw Data Final**



When using GMM, however, we notice a difference in the overall accuracy for raw data and PCA-modified data. GMM classifies the raw data with a 93.99% accuracy, and the PCA-modified one with around 95.99% accuracy. As will be discussed in a later section, we also notice that if we were to further look into the individual cluster groups, the K-means algorithm seems to more accurately classify Benign clusters while the GMM algorithm predicted the positive cases more accurately. For GMM, multiple covariance types were experimented with, giving us the following overall accuracies (Table 2). Our GMM models' accuracies were also quite high. Thus, our models performed well within our expectations. We first tested out which covariance matrix was best for our dataset. The "spherical covariance" worked best for our data with 95.99% accuracy as we compared them visually (Appendix figure 1). This makes sense because spherical covariance works best with circular data that is both axis-aligned and has equal covariance.

Table 2. Table showing the raw results of number of correct labels for each covariance

'Tied'		'Full'		'Diag'		'Spherical'	
PCA	Raw	PCA	Raw	PCA	Raw	PCA	Raw
669	658	669	610	669	652	671	657

From Figure 3, we see that there are clear clusters but the clusters, especially for benign, do not resemble the shape of ellipses that are symbolic of the GMMs. However, we see that the malignant data is better clustered as the data points resemble more of an elongated ellipses while the benign data does not on both raw and PCA data. This is also shown in the Accuracy Table where True Malignant was at 99.59% accuracy versus True Benign was at 94.10% accuracy (Table 1). We discovered that the GMM performed slightly better using the PCA data as there is less overlap between the orange and blue dots than in the raw data.

Overall, we believed that the GMM worked well on our data as it is successful in distinguishing the two different clusters consistently. An interesting point to notice was that we had to perform additional checks to ensure our accuracies were correctly determined. Sometimes, we would get an accuracy of around 5% just because what we assumed to be the benign cluster in our results, was actually the malignant one. Unsupervised machine learning techniques do not make use of pre-defined outcome labels. Instead, they just assess each observation and cluster together data points with similar features. We then cross-checked the cluster membership of each observation with our original data to understand which cluster represented each of our labels - 'Benign' and 'Malignant'.

Discussion -

Our Gaussian Mixture Model had slightly better accuracy working with the PCA data (95.99%) compared to the raw 9-dimensional data (93.99%). We expected there to be a similar trend with the K-means algorithm but were surprised to find the exact same results with both K-means on the raw data and on the PCA data (both 95.7% accurate). Something we could possibly look at if we had more time would be to compare K-means and GMM on PCA data with

a larger range of principal components used rather than only observing the top three principal components.

Gaussian Mixture Models also have the capability to account for non-circular clusters and compare the distances of each point to all the cluster centers rather than just the closest ones. Our comparison of the different covariance types resulted with the 'spherical' option with the highest accuracy. This was interesting since the K-means algorithm spherical covariance type because K-means assumes spherical covariance, but this is likely due to the fact that the actual clusters are generally circular. Because our data was both diagonal and has equal covariance on each axis, giving it a circular shape, we would expect GMM to have a higher accuracy as compared to the K-means algorithm.

Additionally, our GMM models were better at classifying malignant tumors and worse at classifying benign tumors. This is likely due to the elongated forms of the malignant data and more circular forms of the benign data. GMM models work similarly to K-means but since it assumes variables as a mix of Gaussian distributions, it's able to differentiate between overlapping and noisy clusters. By classifying more malignant tumors accurately, it likely resulted in additional false positives and lowered the benign classification accuracy. This is due to the fact that there is a higher likelihood, or a higher value of prior probability (π_k), for malignant labels so there is a higher likelihood for data points to be classified as malignant.

On the other hand, the K-means algorithm had the opposite result in that it had a higher accuracy when classifying benign tumors, as mentioned in results. This could be because of the fact that the data points classified as benign formed a denser cluster with lower variance and more defined boundaries, closer to the 'circular' cluster shapes that K-means works with. On the other hand, the data points associated with malignant tumors formed a larger, less uniformly shaped cluster with softer boundaries which is better suited to GMM algorithms that account for elongation and covariance.

Since our dataset consists of a larger proportion of benign tumors (458:241), if we had more time we would try to handle these imbalanced classes by either looking at different performance metrics, oversampling our minority (malignant) class, or undersampling our majority (benign) class. This may help resolve the difference in performance of our models on benign vs malignant tumors. In addition, we could also take advantage of the probabilistic clustering feature from the Gaussian Mixture Models. Rather than just predicting the cluster labels, we could calculate for the probabilistic cluster assignments and change the threshold to assigning labels to account for the unbalanced dataset. We would also try to run our models on new data outside of our current dataset to analyze the generalizability of our model. This would be an obvious next step considering our ultimate goal is to be able to help doctors prioritize care through early detection of Breast Cancer.

Contributions-

Dina Dehaini: coded the K-means and PCA, comments, on the report - k-means and PCA in methods/appendix

Gauri Samith: coded the K-means and PCA, on the report- Related work and Results

Aditi Krishnakumar: coded the K-means and PCA, on the report - discussion

Grace Gao: coded the GMM, confusion matrices, on the report - discussion

Emilia Pokta: coded the GMM, on the report - results, graphs

Simran Nayyar: coded the GMM, on the report - intro and GMM methods

Code: https://github.com/dinadehaini/COGS-118B-Final-Project

References-

Mohammed, S. A., Darrab, S., Noaman, S. A., & Saake, G. (2020). Analysis of Breast Cancer Detection Using Different Machine Learning Techniques. *Data Mining and Big Data: 5th International Conference, DMBD 2020, Belgrade, Serbia, July 14–20, 2020, Proceedings*, *1234*, 108–117. https://doi.org/10.1007/978-981-15-7205-0_10

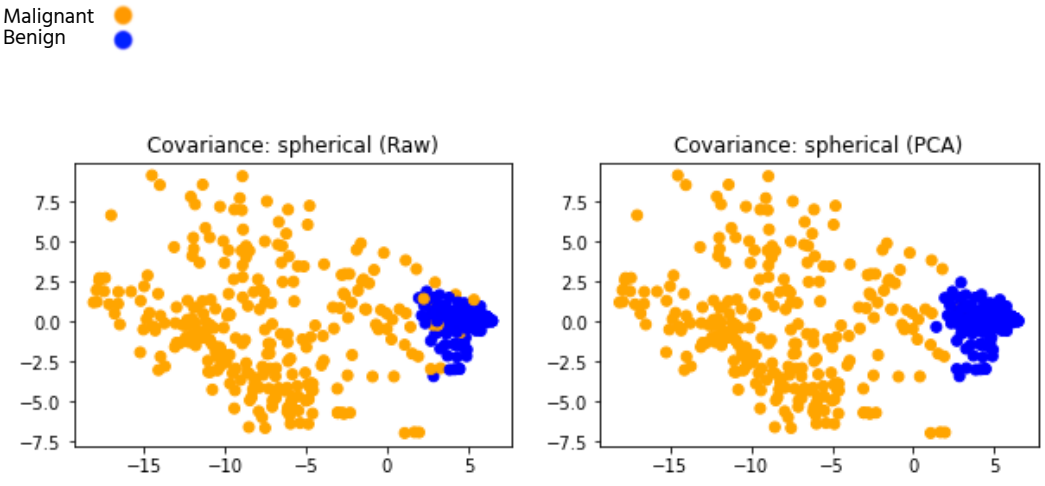
Dubey, A. K., Gupta, U., & Jain, S. (2016). Analysis of k-means clustering approach on the breast cancer Wisconsin dataset. *International journal of computer assisted radiology and surgery*, *11*(11), 2033–2047. https://doi.org/10.1007/s11548-016-1437-9

Appendix-

1. *Overall accuracies*

Covariance Type - Dataset Type	Overall Accuracy
Tied Covariance - PCA Data	95.708%
Tied Covariance - Raw Data	94.13%
Full Covariance - PCA Data	95.708%
Full Covariance - Raw Data	87.26%
Diagonal Covariance - PCA Data	95.708%
Diagonal Covariance - Raw Data	93.27%
Spherical Covariance - PCA Data	95.99%
Spherical Covariance - Raw Data	93.99%

2. *Scatter plots of both raw data(left) and PCA data(right) after fitting into Gaussian Mixture Model*

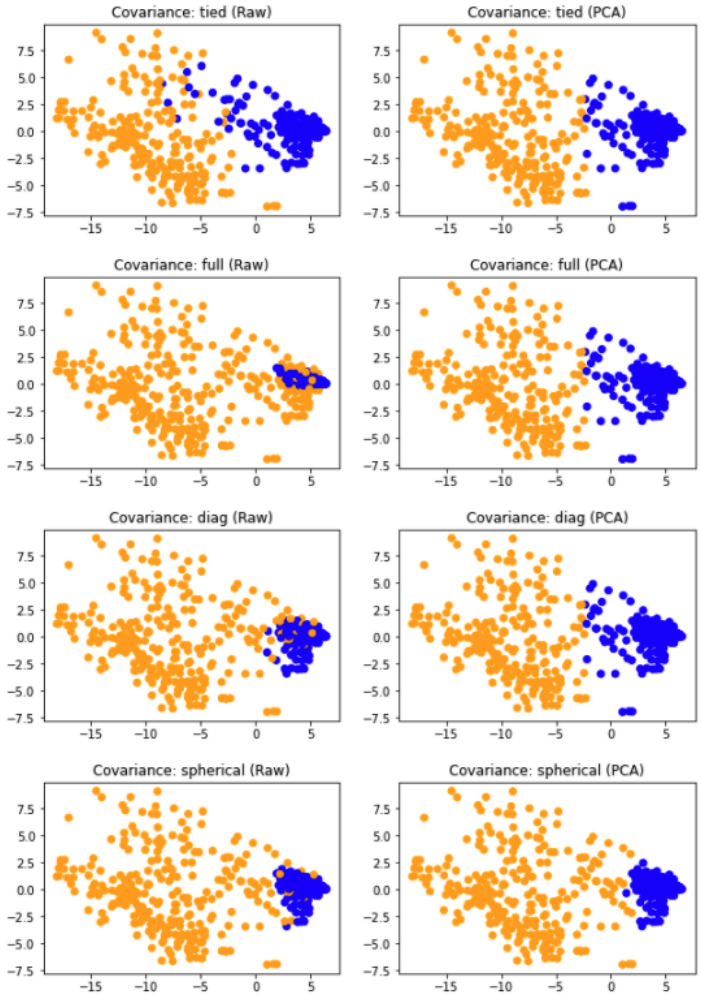


3. *Table showing the raw results of number of correct labels for each covariance*

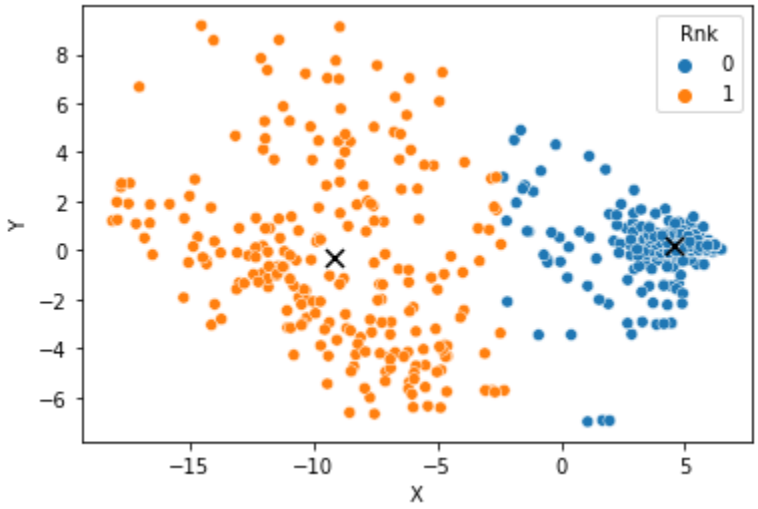
'Tied'		'Full'		'Diag'		**'Spherical'**	
PCA	Raw	PCA	Raw	PCA	Raw	**PCA**	Raw
669	658	669	610	669	652	**671**	657

4. *Scatter plots of GMM with different covariance matrices and varying raw and PCA data*

```
tied pca matches: 669
tied raw matches: 658
full pca matches: 669
full raw matches: 610
diag pca matches: 669
diag raw matches: 652
spherical pca matches: 671
spherical raw matches: 657
```



5. *2D PCA raw and PCA final cluster*



6. *Final cluster assignments compared to actual cluster assignments*