IMPLEMENTATION OF KMEANS ON IRIS AND IMAGE DATA USING DIFFERENT DIMENSIONALITY REDUCTION TECHNIQUES

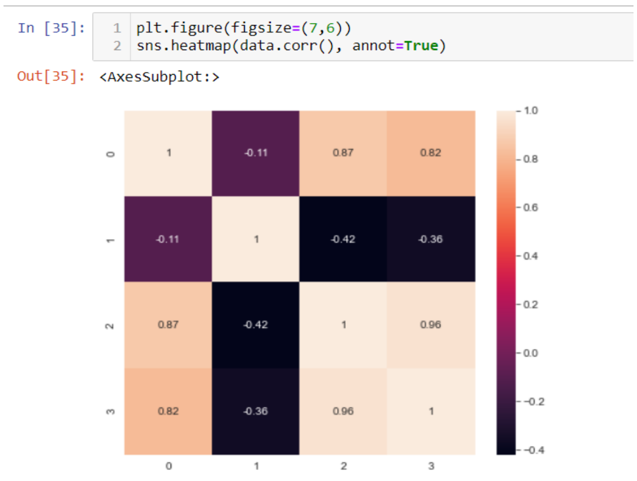
Username: rautaniket18

Accuracy on Iris dataset: 95% **Rank:** 81

Accuracy on MNIST Image dataset: 67% **Rank**: 581

Approach:

➢ Applying K-means for clustering on Iris data
The given iris dataset is small enough with the size of 150*4 odd entries.
As per the below heat map, it is clearly evident that out of 4 columns, only 2 columns are related.



```
In [35]:  plt.figure(figsize=(7,6))
          sns.heatmap(data.corr(), annot=True)

Out[35]: <AxesSubplot:>
```

The code defines a class Kmeans_Clustering that takes the number of clusters, iterations, and a random seed as input. The class contains several methods that implement the different steps of the K-means algorithm, including initializing the centroids, calculating the distances, assigning data points to clusters, updating centroids, and computing the sum of squared errors. The algorithm creates empty clusters and assigns each data point to the closest centroid using Euclidean distance. It repeats this process for a specified number of iterations, updating the centroids to be the mean of each cluster's data points and calculating the SSE for each cluster. If the new centroids are close enough to the old centroids, the algorithm stops. Otherwise, the process continues with the new centroids.

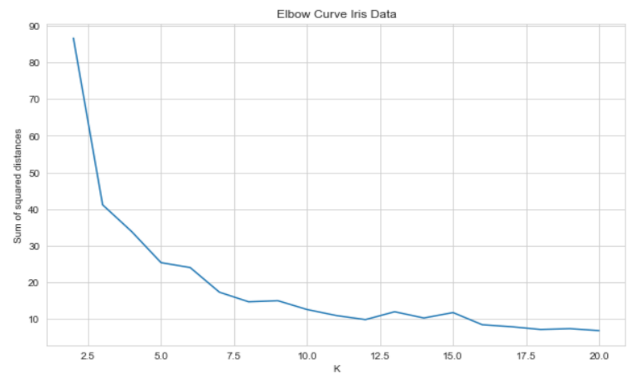
I applied the Kmeans directly to the data and was able to get very low accuracy compared to that of the scaled data. I applied normalization, standard scalar, applied PCA or T-SNE, and increased the iterations. After increasing the iterations and applying got to see a huge difference in accuracy and got the best accuracy of 90% to 95%. Using PCA got 90% whereas using T-SNE got 95%. Which suggests that T-SNE performs slightly better.

Model	Iterations	Cluster	Normalize	Std. Scalar	Accuracy
Model with T-SNE	100	3	Y	Y	65
	300	3	Y	Y	90
	3000	3	Y	Y	95

Model	Iterations	Cluster	Normalize	Std. Scalar	Accuracy
Model with PCA	50	3	Y	Y	65
	100	3	Y	Y	61
	300	3	Y	Y	67
	3000	3	Y	Y	90

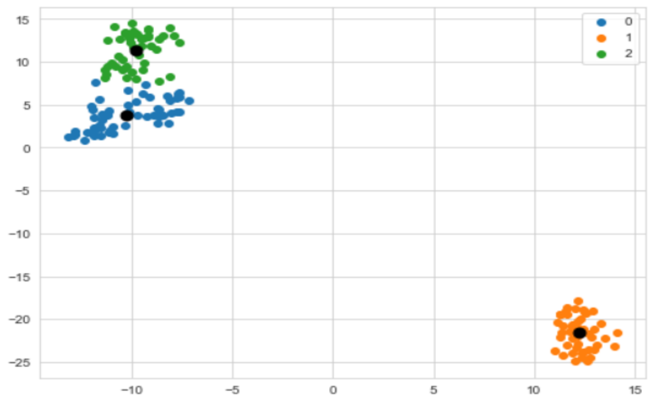
Model	Iterations	Cluster	Normalize	Std. Scalar	Accuracy
Sk-Learn Kmeans	3000	3	Y	Y	95%

From the above observations, it is clear that as iterations increase, accuracy increases irrespective of the PCA / T-SNE. When I compared my best model (using T-SNE) with the inbuilt sk-learn library (using T-SNE) both of them performed the same with 95% accuracy.



A good model means a model with low SSE and a low number of clusters. To find the best optimal K, the elbow method is used which selects the point where SSE begins to ground. From the above graph, at the start the SSE decreases drastically, and then after 5 there is a minimal decrease. At K=3 it is evident that there is a slight elbow.

Below is the scatter plot for K=3 with its centroids.



- Applying K-means for clustering on Image data
  The given Image dataset has a size of 10000*784, which means the data has a large number of features that can increase the multicollinearity, risk of overfitting, large storage, and hard to visualize. To overcome this problem, I have used dimensionality reduction techniques such as PCA and T-SNE.
  I have Normalized the data which scaled the data to a common scale, as it applies to rows, and has applied standard scalar which applies to columns.

Model	Iterations	Cluster	Components	Normalize	Std. Scalar	Accuracy
Kmeans with PCA	100	10	50	Y	Y	43
	200	10	50	Y	Y	45
	300	10	50	Y	Y	45

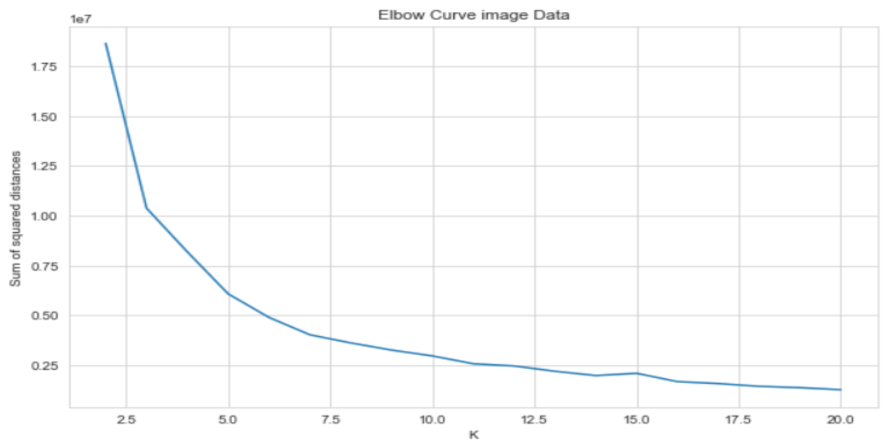
Model	Iterations	Cluster	Components	Normalize	Std. Scalar	Accuracy
Kmeans with PCA	100	10	100	Y	Y	43
	200	10	100	Y	Y	49
	300	10	100	Y	Y	46

Model	Iterations	Cluster	Components	Normalize	Std. Scalar	Accuracy
Kmeans with PCA	100	10	150	Y	Y	44
	200	10	150	Y	Y	45
	300	10	150	Y	Y	44

Model	Iterations	Cluster	Components	Normalize	Std. Scalar	Accuracy
Kmeans with T-SNE	100	10	2	Y	Y	67
	100	10	3	Y	Y	67

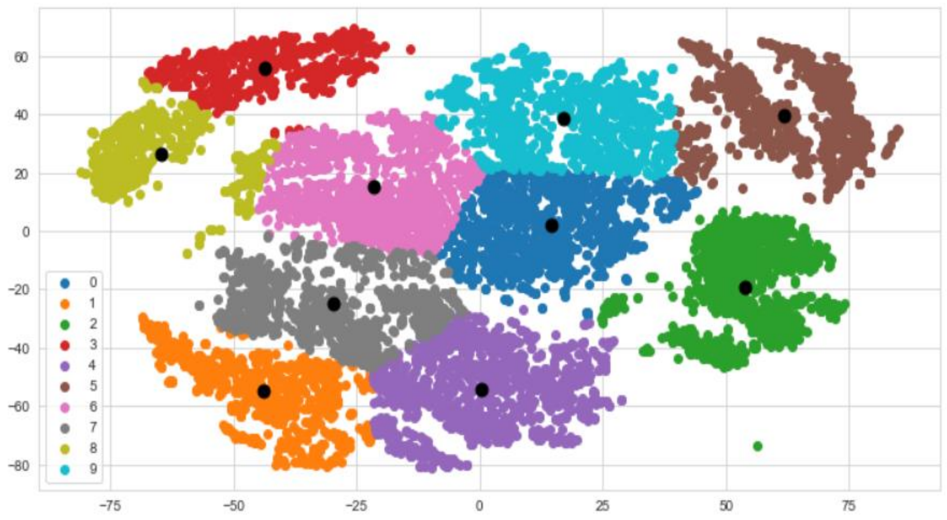
Model	Iterations	Cluster	Components	Normalize	Std. Scalar	Accuracy
Sk-Learn Kmeans	100	10	2	Y	Y	67

From the above-listed data, it is clear that the accuracy of the model is between 43% to 46% when used with PCA with different combinations of components and iterations.
When the same model is used with the T-SNE reduction technique it gives a better result with 67% on 100 iterations and either 2 or 3 components.
From both the iris and image dataset technical analysis it is visible that T-SNE has the upper hand over PCA.
The same data obtained after T-SNE was passed to sk-learn Kmeans and the accuracy is 67% which is the same, compared with the Kmeans (from scratch with T-SNE).



The sum of squared error reduces with an increase in the number of clusters (K). This is so that the data points may be more accurately fitted to their corresponding clusters, which is made possible by having more clusters.
As K rises, the sum of squared error loss, however, reaches a limit at which it begins to decline. This point in the graph, which is close to K=10, indicates that there might be a good number of clusters at this point.

Below is the scatter plot for K=10 with its centroids.



Conclusion:
In conclusion, I implemented K-means clustering on Iris and image datasets using different dimensionality reduction techniques such as PCA and T-SNE. I observed that normalization, standard scaling, and increasing the iterations improved the accuracy of the K-means model. I also used the elbow method to find the optimal number of clusters and found that for the Iris dataset, K=3 was the best number of clusters, while for the image dataset, K was around 10. The accuracy of the KMeans algorithm on the image dataset is lower compared to the iris dataset due to the higher complexity and variability of the data. T-SNE performed better than PCA in both datasets. Overall, the results suggest that T-SNE is a better choice for dimensionality reduction when performing K-means clustering on high-dimensional datasets.