Implementing & Analysing Gaussian Mixture Models in LIME

Name: Satyam Arora (2020330)

Data: December 2023

1 Introduction

1.1 Gaussian Mixture Models

Gaussian Mixture Models (GMMs) are a probabilistic model for representing normally distributed subpopulations within an overall population. In machine learning, GMMs are used for clustering applications, where the goal is to identify subgroups within a dataset. A GMM assumes that the data is generated from a mixture of several Gaussian distributions, each with its own mean and covariance. One of the key features of GMMs is their capability to model complex distributions through a combination of these simpler Gaussian distributions.

Unlike K-means clustering, which assigns each data point to a single cluster, GMMs assign a probability (or weight) indicating the likelihood of a data point belonging to each cluster. This makes GMMs a soft clustering technique, providing more information about the uncertainty of the clustering process. The Expectation-Maximization (EM) algorithm is typically used to estimate the GMM parameters, which involves iteratively updating the cluster assignments and the Gaussian parameters to maximize the likelihood of the data given the model.

GMMs are particularly useful in scenarios where the data is assumed to come from several overlapping Gaussian distributions, and you need a model that can capture this complexity. They are widely used in various fields, including image processing, pattern recognition, and bioinformatics, for tasks such as image segmentation, anomaly detection, and clustering analysis. The flexibility and probabilistic nature of GMMs make them a powerful tool for understanding and extracting meaningful patterns from complex datasets.

1.2 Local Interpretable Model-agnostic Explanations

LIME is a novel technique designed to explain the predictions of any machine learning classifier in an interpretable and faithful manner. By approximating

the model locally with an interpretable model, LIME helps to demystify complex black-box models such as deep neural networks or ensemble methods.

It works by generating a new dataset consisting of perturbed samples around a specific instance, then learning a simple, interpretable model (like a linear model) on this new dataset. The interpretable model aims to be locally faithful, meaning it should approximate the original model's predictions well near the explained instance. LIME's ability to provide insights into individual predictions makes it an invaluable tool for understanding and trusting machine learning models, particularly in scenarios where interpretability is as crucial as predictive accuracy.

1.3 Current Goal

Since LIME uses what we can call a "Surrogate" Model, which can be any model with a definite distribution, to generate explanations, we can use GMM. GMMs when implemented, have closed-form solutions to their parameters, so we can easily sample from them and do not need to calculate a posterior distribution ourselves.

In this report, I have covered the whole journey, starting from Classical Machine Learning (Data preprocessing, then EDA along with Simple Model Training) to implementing K-Means, to Dimensionality Reduction (using both PCA & TSNE), then to GMMS (for all 4 covariance types), and finally to LIME explanations generated through these GMMS. Throughout the report, I plotted various graphs to visually understand every technique and calculated results at every step.

2 Using Traditional Machine Learning

2.1 About the Dataset

I have used the `Red Wine Quality` Dataset from the UCI Machine Learning Repository Link.

Feature Information:

Variable Name	Role	Type	Demographic	Description	Units	Missing Values
fixed_acidity	Feature	Continuous				no
volatile_acidity	Feature	Continuous				no
citric_acid	Feature	Continuous				no
residual_sugar	Feature	Continuous				no
chlorides	Feature	Continuous				no
free_sulfur_dioxide	Feature	Continuous				no
total_sulfur_dioxide	Feature	Continuous				no
density	Feature	Continuous				no
pH	Feature	Continuous				no
sulphates	Feature	Continuous				no

There are 11 features, all of them continuous with 0 null values.

Target variable: Quality, it has 6 unique categories [3, 4, 5, 6, 7, 8].

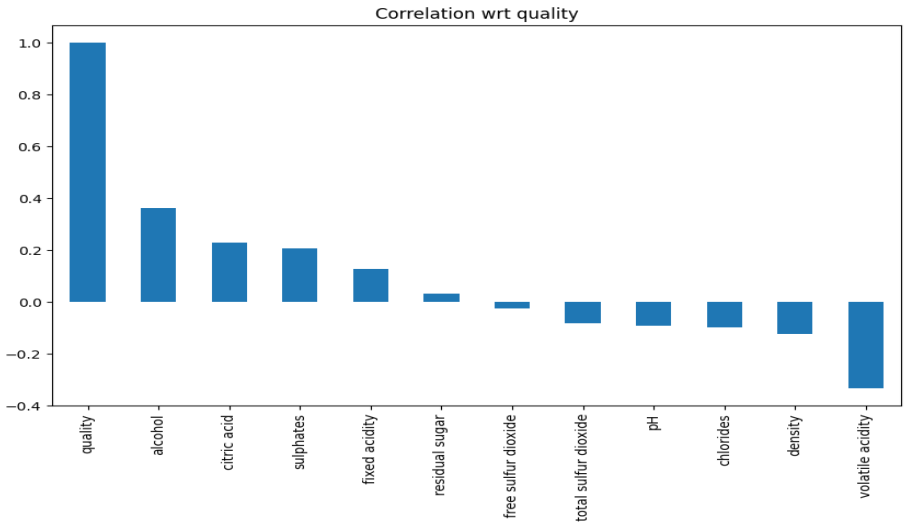
$$\text{Quality} = \begin{cases} \text{low} & \text{if Quality} < 5 \\ \text{medium} & \text{if } 5 \leq \text{Quality} < 7 \\ \text{high} & \text{if Quality} \geq 7 \end{cases}$$

Final counts: Low = 63, medium = 1319, high = 217

I use StandardScalar() to scale the dataset, then use RandomForestClassifier with parameters n_estimators = 100 & criterion = "gini".

	precision	recall	f1-score	support
high	0.77	0.65	0.71	37
low	0.33	0.07	0.11	15
medium	0.91	0.97	0.94	268
accuracy			0.89	320
macro avg	0.67	0.56	0.58	320
weighted avg	0.86	0.89	0.87	320

As you can see, this model gives an accuracy of 90 (okay, considering we have plainly fitted the model with data augmentation techniques, etc.)



Feature Importance Using RandomForestClassifier:

1. fixed acidity: 0.060892
2. volatile acidity: 0.066550
3. citric acid: 0.074610
4. residual sugar: 0.076734
5. chlorides: 0.080768
6. free sulfur dioxide: 0.089230
7. total sulfur dioxide: 0.092655
8. density: 0.094481
9. pH: 0.109310
10. sulphates: 0.120002
11. alcohol: 0.134769

Most of the features are equally important.

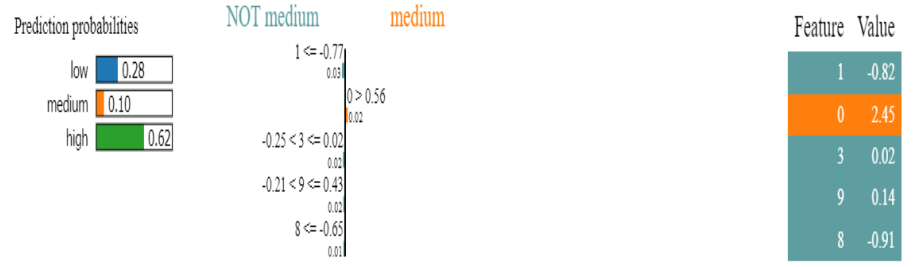
Now using PCA with n_components = 4 and applying the same RandomForestClassifier with the same parameters, we get 87 as the Accuracy.

Feature Importance after Using PCA:

1. Feature 1: 0.282795
2. Feature 2: 0.174531
3. Feature 3: 0.144488
4. Feature 4: 0.111410

Using TSNE with n_components = 3 with the same classifier gives 83 as the classification score.

At this point, we can use LIME to generate explanations, one such example is given below:



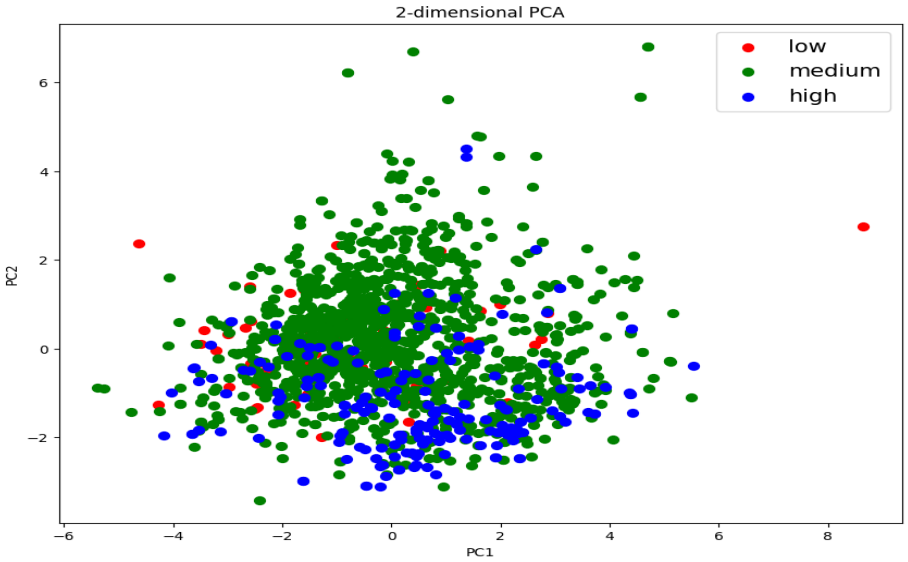
The LimeTabularExplainer is used, it creates interpretable machine-learning models for tabular data. It takes the training data as input and labels each class ('low', 'medium', 'high'). The explainer will then generate local explanations for predictions, focusing on how different features in the data influence the prediction for individual instances.

In the image, Prediction probabilities for the 3 classes with respect to the target input are shown, along with how the top-5 features affect the decision, in tabular and graphic format. LIME currently can't depict the multi-class-wise effect of features. Therefore, it uses the binary method of Not Medium Vs Medium Classification.

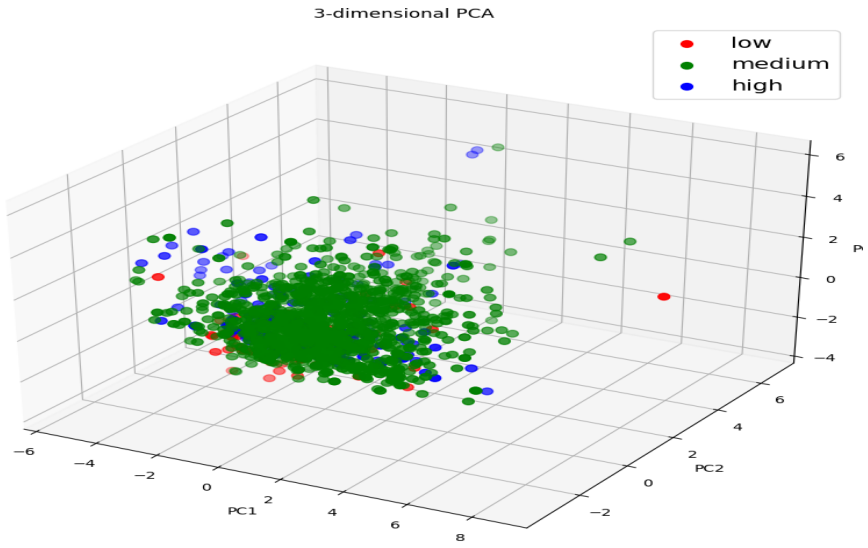
3 Applying Regular Clustering

On the Scaled Features, we can apply several dimensionality reduction techniques, I have applied PCA and TSNE both (with 2 and 3 components each).

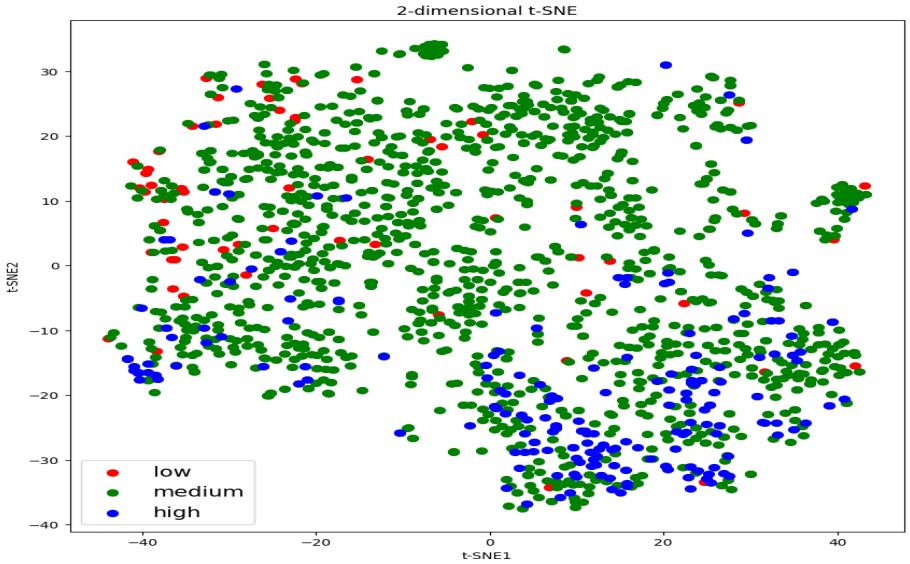
3.1 PCA with 2 Components



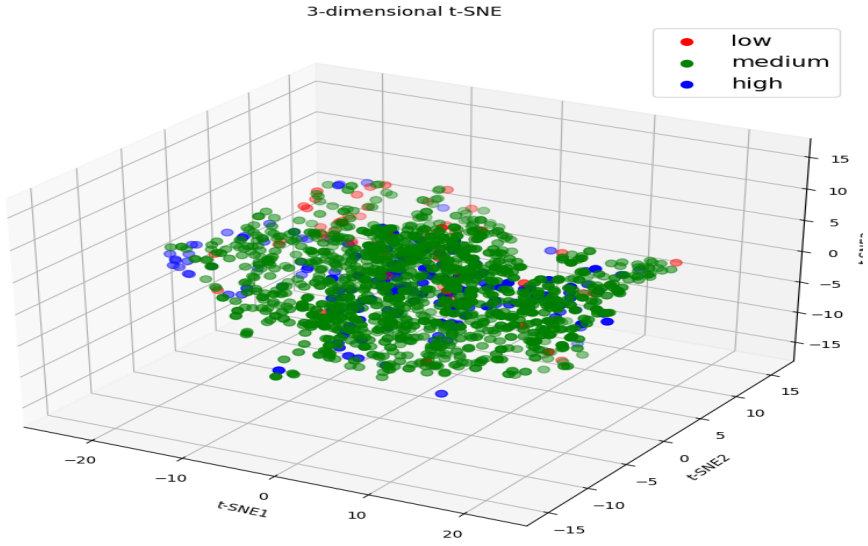
3.2 PCA with 3 Components



3.3 TSNE with 2 Components



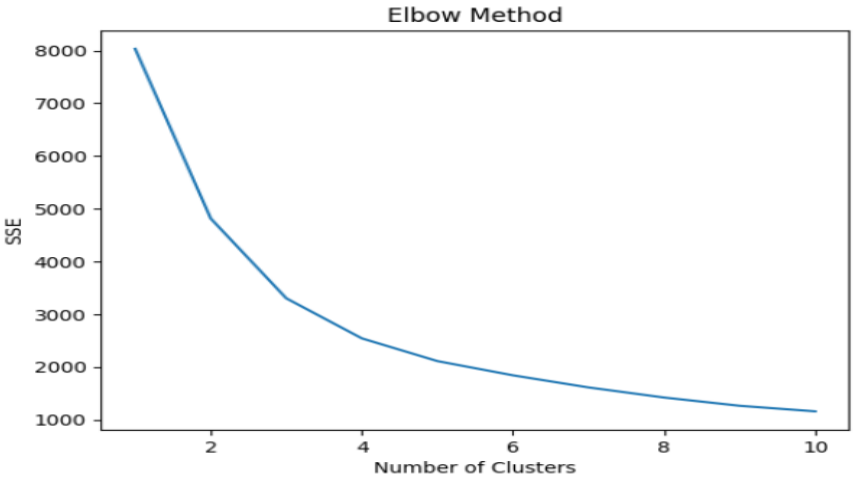
3.4 TSNE with 3 Components



From the plots, it is clear that Simple Clustering would not yield good results. However, it can still be used as a measure to compare the accuracy of GMMs later on.

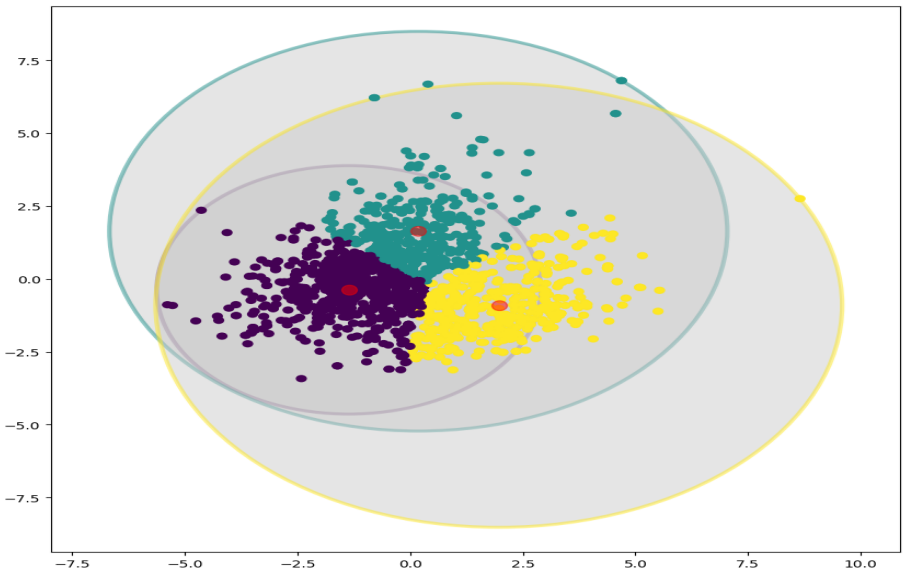
3.5 K-Means and its results

First, let us calculate the Elbow or the optimal number of Clusters required for the Dataset. By fitting from 1 to 11, we get the following results:



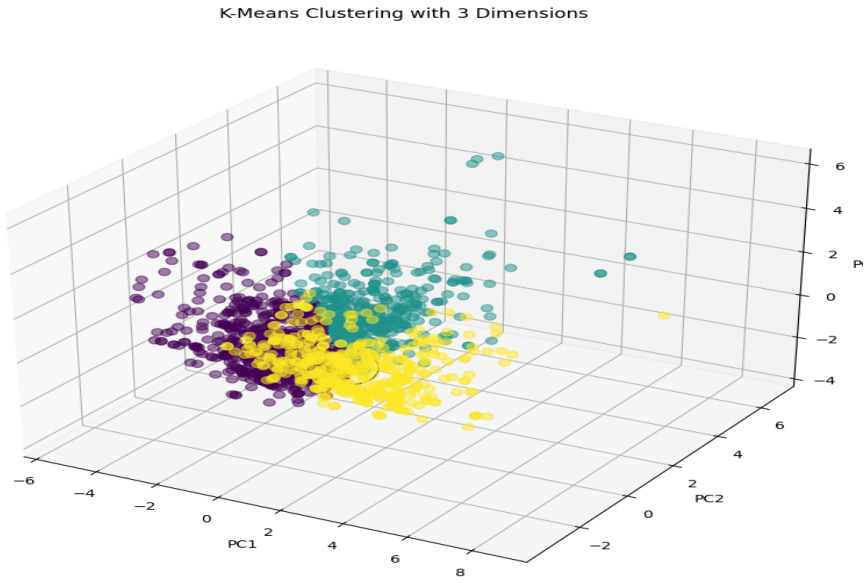
From this graph, we can clearly see that we can use K = 3 as an optimal number of clusters.

3.5.1 K-Means on PCA with 2 components



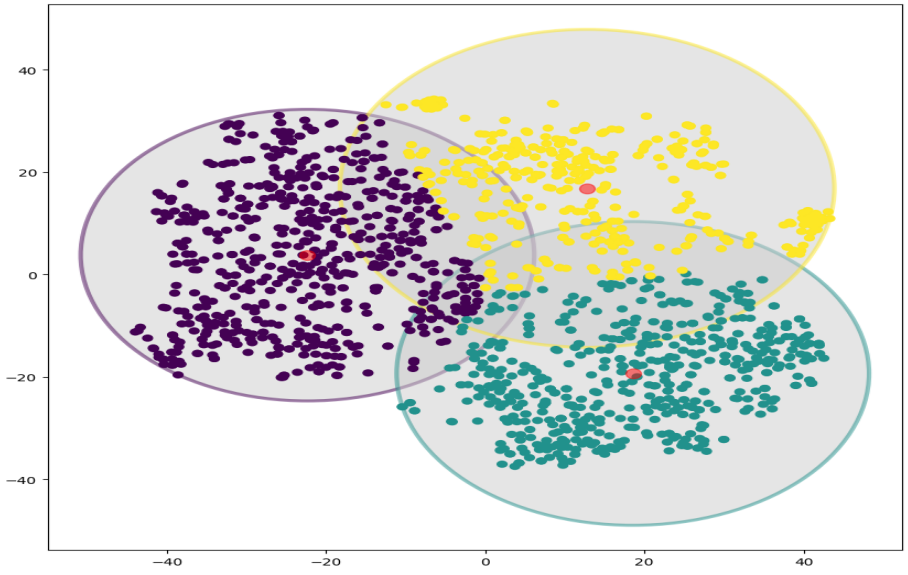
The accuracy score we get is: 0.3452157598499062

3.5.2 K-Means on PCA with 3 components



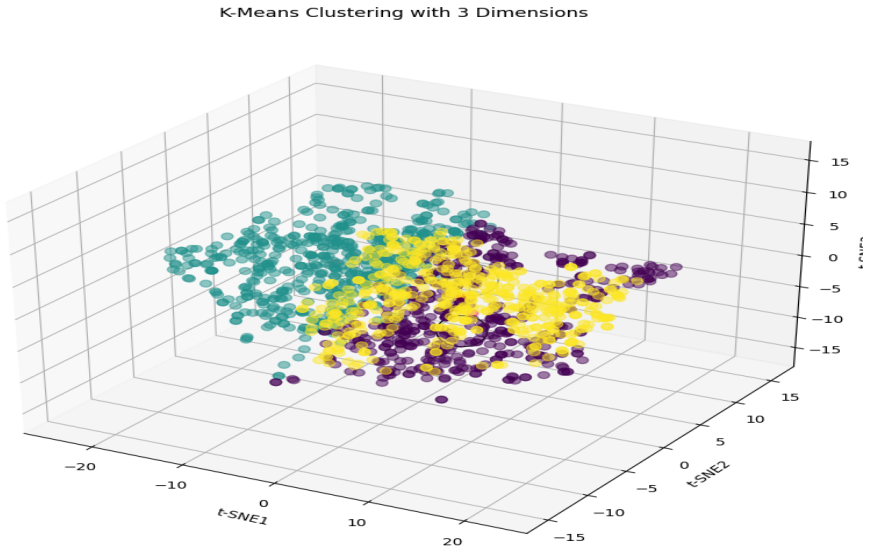
The accuracy score we get is: 0.32770481550969355

3.5.3 K-Means on TSNE with 2 components



The accuracy score we get is: 0.2614133833646029

3.5.4 K-Means on TSNE with 3 components



The accuracy score we get is: 0.41088180112570355

After Applying Dimensionality Reduction, the best accuracy we could get is 41, given by TSNE with 3 components.

Reasons for lower accuracy than the original:

- The dataset Clusters are not independent, therefore simply applying clustering is catastrophic.
- RFC on original depicted that feature importance is distributed almost equally in most of the features, therefore clubbing them together further worsens the accuracy.

Reasons for better accuracy than its peers:

- TSNE tends to save bits of the original data structure, which can be seen through the plot.
- Increased number of components also helps in distributing feature importance.

4 Implementing Gaussian Mixture Models

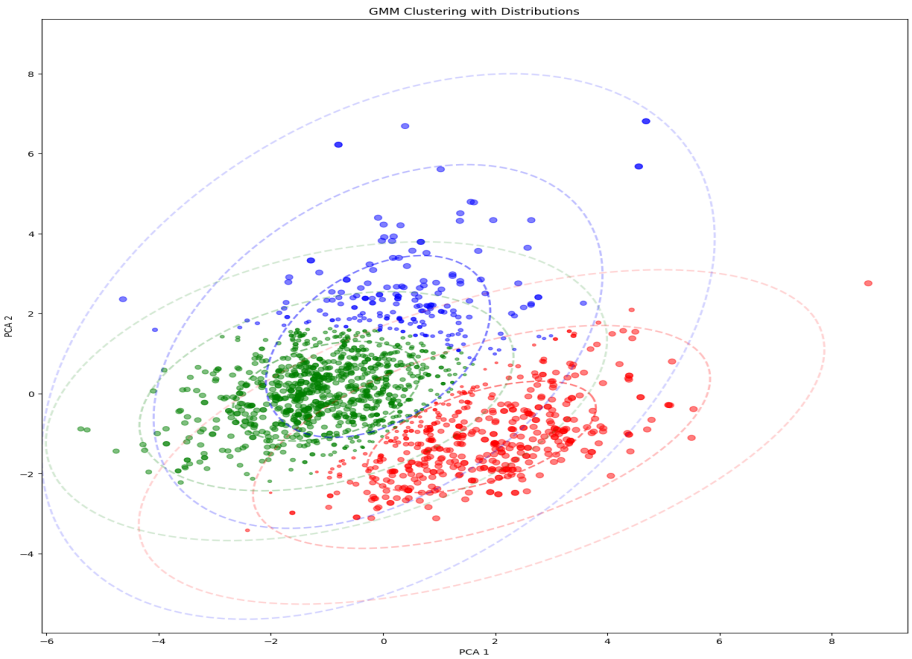
Now, using the features obtained from PCA and TSNE, we can apply GMMs. We will be having the same cluster size K as K-Means (3).

GMMs have 4 Covariance Types:

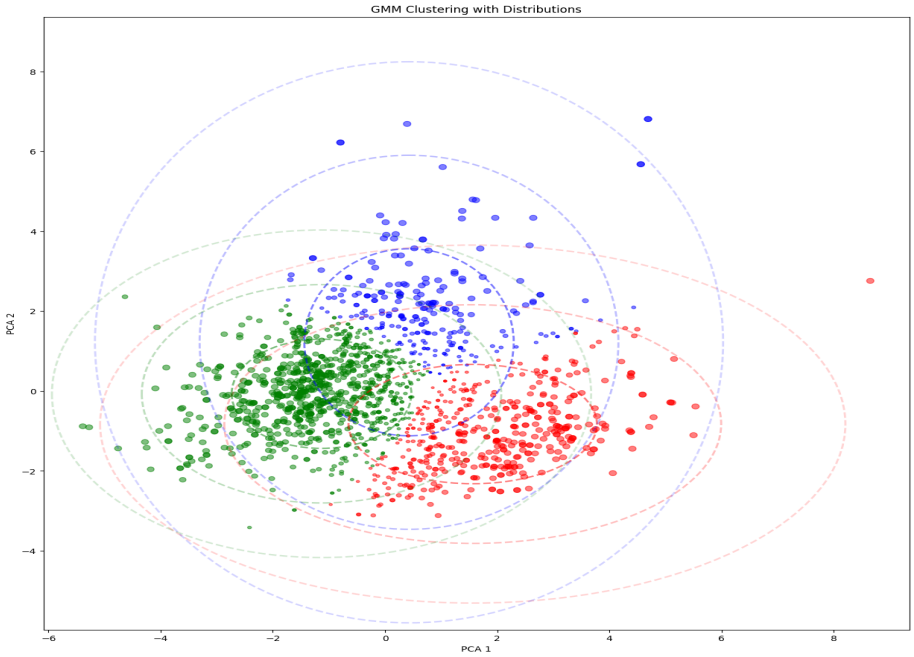
- full: Each component has its own general covariance matrix, allowing for ellipsoidal shapes in any orientation.
- diag: All components share the same general covariance matrix, resulting in similarly shaped and oriented ellipsoids.
- tied: Each component has its own diagonal covariance matrix, implying ellipsoids that can be oriented along the coordinate axes but may differ in size.
- spherical: Each component has its own single variance, resulting in spherical clusters that are equal in all directions.

In each of the plots below, the points are scattered based on their Prediction Probabilities. For points closer to multiple clusters, prediction probabilities would be distributed, meaning low confidence, and the size of such points would be lesser than those with confidence within the cluster.

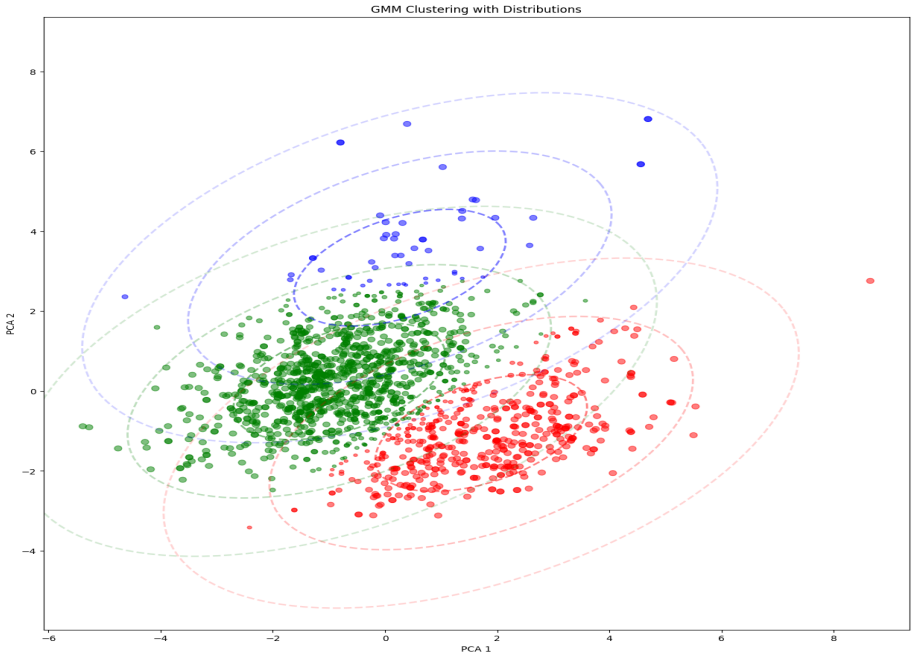
4.0.1 GMM with full covariance on PCA with 2 components



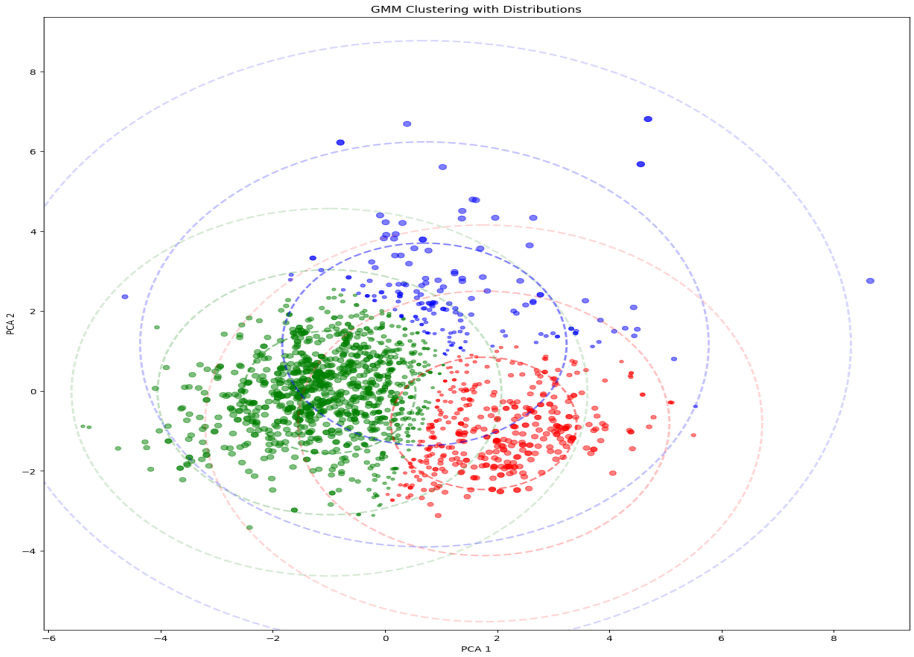
4.0.2 GMM with diag covariance on PCA with 2 components



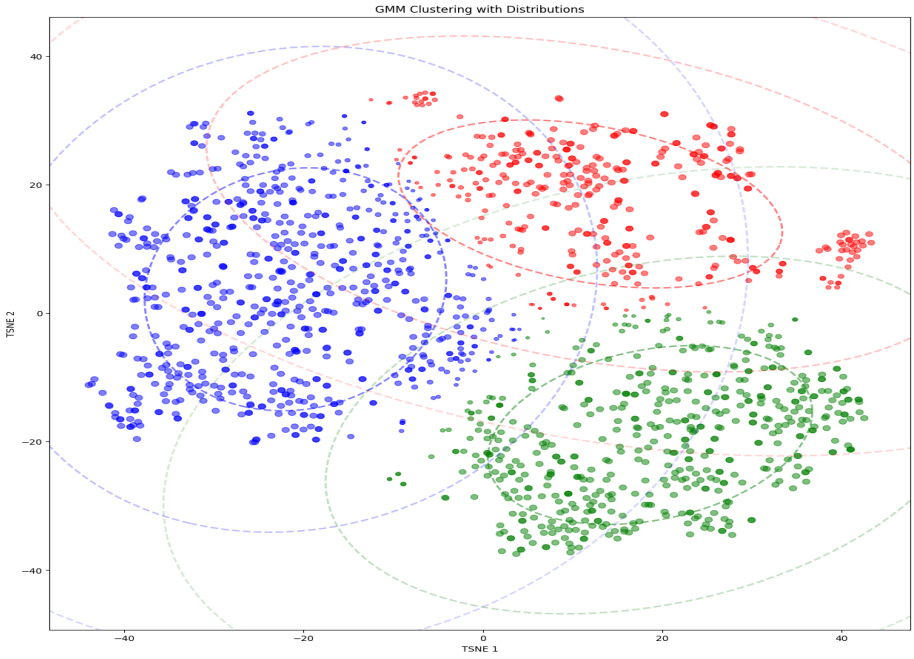
4.0.3 GMM with tied covariance on PCA with 2 components



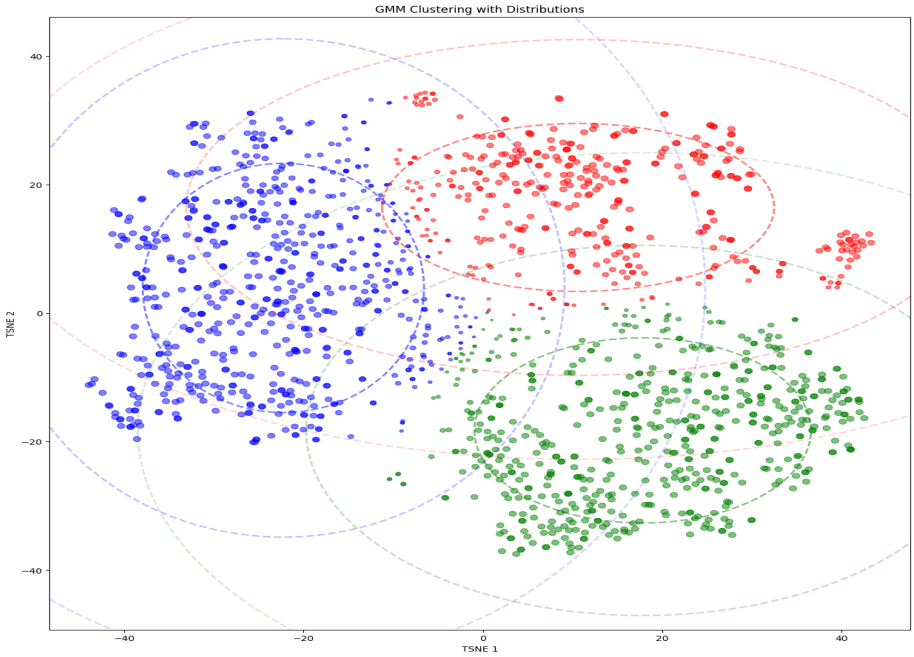
4.0.4 GMM with spherical covariance on PCA with 2 components



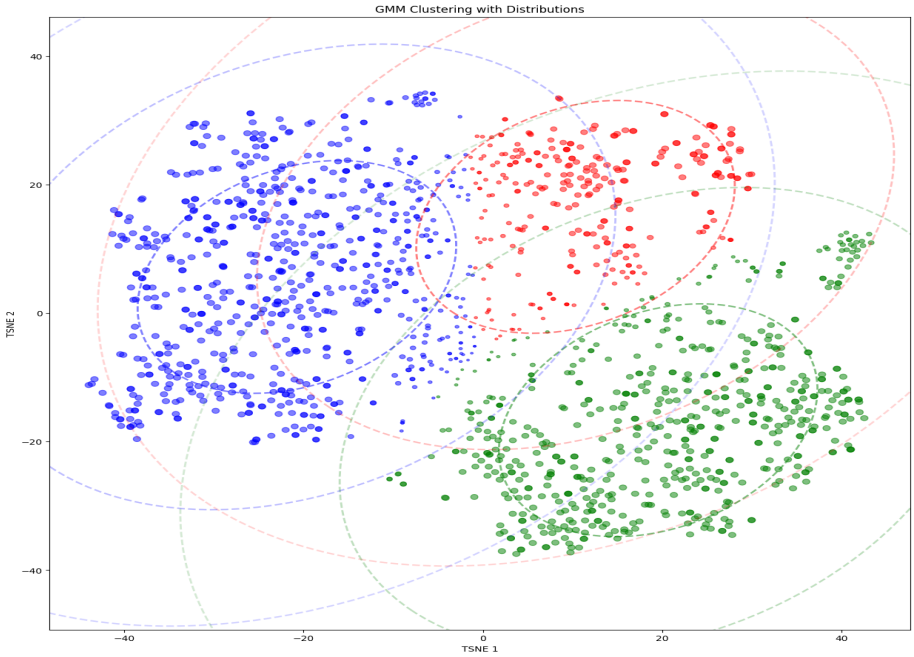
4.0.5 GMM with full covariance on TSNE with 2 components



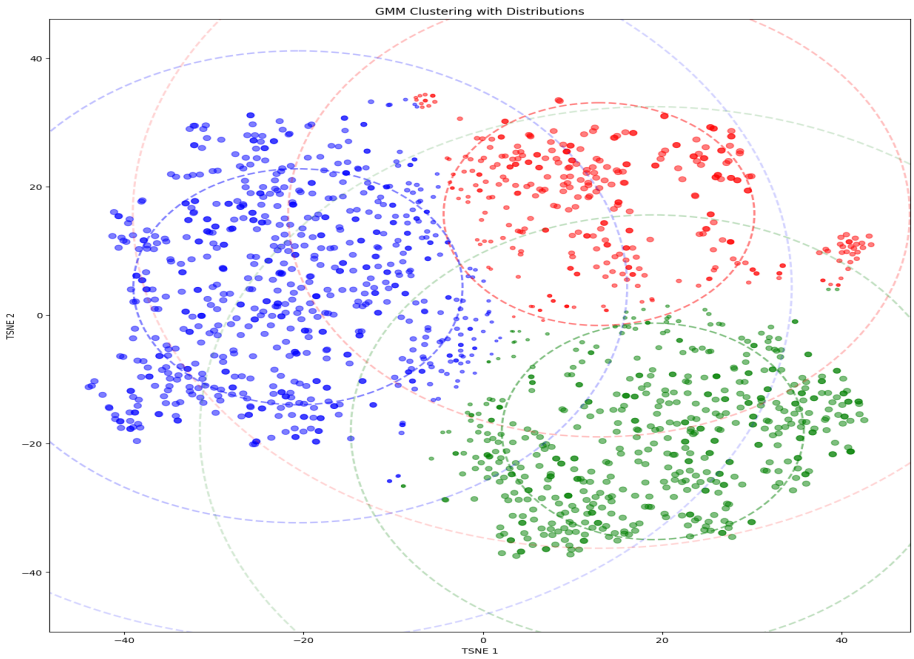
4.0.6 GMM with diag covariance on TSNE with 2 components



4.0.7 GMM with tied covariance on TSNE with 2 components



4.0.8 GMM with spherical covariance on TSNE with 2 components



Through these visualizations, we can see how the covariance type affects the distribution structure of each cluster.

Their effects on the distribution boundaries are as follows:

- full: Allows each cluster to have an ellipsoidal shape in any orientation and size, offering the most flexibility in capturing the data's distribution.
- diag: Each cluster can have an ellipsoidal shape, but the axes of the ellipses are aligned with the coordinate axes. This leads to clusters stretched along the axes.
- tied: Forces all clusters to have the same ellipsoidal shape and orientation, but they can be located differently. This results in parallel ellipses in 2D.
- spherical: Restricts clusters to spherical shapes, meaning they are circular in 2D and do not have any orientation.

5 Final Results along with Explanations

5.1 Accuracy of GMM on Scaled Features:

- full: 0.58
- diag: 0.60
- tied: 0.70
- spherical: 0.55

Classification Report and LIME explanation of best model:

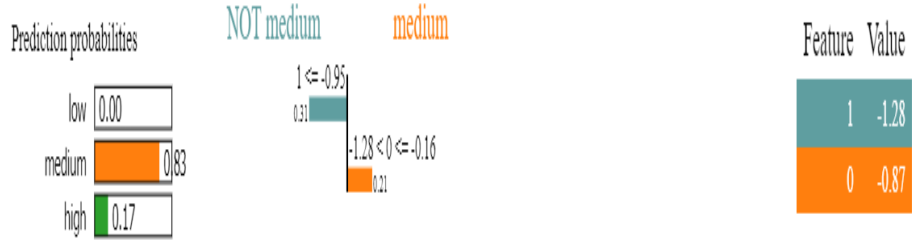
	precision	recall	f1-score	support
0	0.06	0.03	0.04	63
1	0.89	0.73	0.81	1319
2	0.31	0.70	0.43	217
accuracy			0.70	1599
macro avg	0.42	0.49	0.43	1599
weighted avg	0.78	0.70	0.72	1599

5.2 Accuracy of GMM on PCA with 2 Components:

- full: 0.58
- diag: 0.56
- tied: 0.67
- spherical: 0.61

Classification Report and LIME explanation of best model:

	precision	recall	f1-score	support
0	0.02	0.02	0.02	63
1	0.87	0.71	0.78	1319
2	0.28	0.60	0.38	217
accuracy			0.67	1599
macro avg	0.39	0.44	0.39	1599
weighted avg	0.76	0.67	0.70	1599

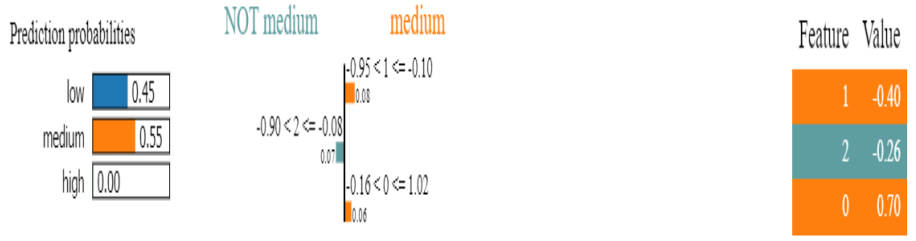


5.3 Accuracy of GMM on PCA with 3 Components:

- full: 0.56
- diag: 0.43
- tied: 0.62
- spherical: 0.55

Classification Report and LIME explanation of best model:

	precision	recall	f1-score	support
0	0.01	0.03	0.01	63
1	0.90	0.68	0.78	1319
2	0.23	0.41	0.29	217
accuracy			0.62	1599
macro avg	0.38	0.38	0.36	1599
weighted avg	0.78	0.62	0.68	1599

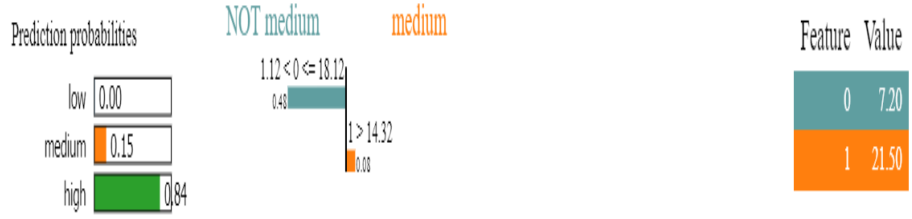


5.4 Accuracy of GMM on TSNE with 2 Components:

- full: 0.49
- diag: 0.46
- tied: 0.50
- spherical: 0.50

Classification Report and LIME explanation of best model:

	precision	recall	f1-score	support
0	0.03	0.11	0.04	63
1	0.88	0.48	0.62	1319
2	0.27	0.76	0.40	217
accuracy			0.50	1599
macro avg	0.39	0.45	0.35	1599
weighted avg	0.76	0.50	0.57	1599

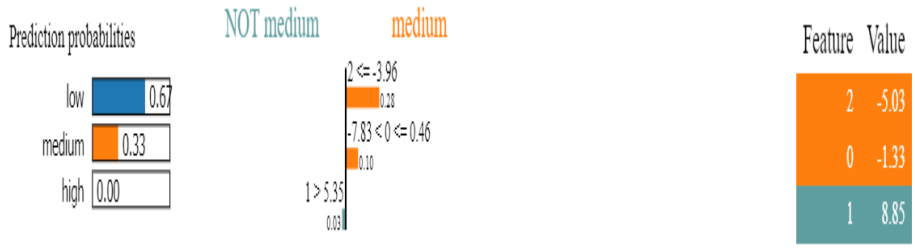


5.5 Accuracy of GMM on TSNE with 3 Components:

- full: 0.44
- diag: 0.43
- tied: 0.43
- spherical: 0.42

Classification Report and LIME explanation of best model:

	precision	recall	f1-score	support
0	0.07	0.65	0.13	63
1	0.94	0.39	0.55	1319
2	0.32	0.71	0.44	217
accuracy			0.44	1599
macro avg	0.44	0.58	0.37	1599
weighted avg	0.82	0.44	0.52	1599



6 Final Conclusion

GMM was performed much better than any other classification model for this dataset. Even though this dataset had dependent clusters, and the Random Forest Classifier trained on complete data got 90% accuracy. Still, the GMM could get 70% accuracy when trained on the complete feature set.

GMMs actual prowess can be seen when the GMM with tied covariance type trained on a meagre 2 component PCA dataset could achieve 67% accuracy. It shows GMMs robustness compared to other models, and with such a robust model, the LIME explanations are bound to be robust too.

GMM could outperform K-Means by almost 200% in accuracy, showcasing that even in a dataset that is not meant for clustering, it could still perform fairly compared to algorithms like RandomForest and much better than algorithms like K-Means.

I have showcased GMMs worthiness, and such a model is bound to be trustworthy, increasing trust in the predictions and explanations.

Thanks and Regards