TASK 2 -MUSIC GENRE ANALYSIS

Introduction:

This aim of the project is to develop a music genre analysis model that classifies songs based on their musical features. The process involved multiple stages including vectorisation using **Bag Of Words**, dimensionality reduction using **PCA(Principal Component Analysis)**,clustering using **K-Means++**, and classification using **KNN(K- nearest neighbours)**.

1. Feature Vectorisation using Bag Of Words:

Since our dataset consisted of textual descriptions of musical characteristics. (add here)There are many famous ways like BoW and TFIdf to do this. TfIdf measures the importance of each word in the document using Tf(Term Frequency) and Idf(Inverse document frequency) scores given by :

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = log\frac{N}{1+df}$$

$$TF-IDF(t,d) = TF(t,d) * IDF(t)$$

Rare words(appearing in fewer documents) get a higher IDF score whereas Common words(appearing in many documents) get a lower IDF score. Using this method will create an inconsistency in our analysis. For example, in the given dataset the keyword ‘Guitar’ appears more frequently than ‘synth’ , but this does not make the word ‘Guitar’ less meaningful or significant by any means.

Hence, I have used the Bag of Words technique for vectorisation .In this method, each unique word becomes a feature (or a dimension) and frequency of each word in a given description was recorded.

2. Dimensionality reduction using PCA(Principal Component Reduction):

The BoW representation results in high- dimensional data , which could be computationally expensive and challenging to interpret. To address this, Principal

Component Analysis(PCA) was applied to reduce the number of dimensions while retaining most of the variance in the data. This step also helps in visualising the music feature space and improving the efficiency of subsequent clustering and classification steps. The set of dimensions found using PCA are orthogonal and linearly independent.

Procedure:

1 . Standardising the dataset:

We first standardise the dataset so that all features have zero mean and unit variance. Mathematically, each feature X is standardised as:

3. Computing the covariance matrix:

The covariance matrix captures relationships between different features. If two features are highly correlated, PCA will try to merge their variance into fewer principal components.

For an n-dimensional dataset, the covariance matrix is an n * n matrix, calculated as:

A high covariance between two features means they provide similar information, and PCA will combine them into one principal component.

4. Compute Eigenvalues and Eigenvectors:

To determine the new feature space ,we compute the eigenvalues and eigenvectors of the covariance matrix.

- Eigenvectors represent the principal components (PCs) (new axes in transformed space)
- Eigenvalues indicate the amount of variance captured by each eigenvector.

5. Select the Principal Components:

We rank the eigenvectors by their eigenvalues (i.e., how much variance they explain) and select the top k values (Here, k = 2) as our new dimensions

3. Combining embeddings into one:

I have used the weighted mean approach to combine the obtained three 2-Dimensional vectors into one. The weights are given based on the variance of each vector (Since, high variance implies high importance and thus high weight).

4.Clustering using K-Means++:

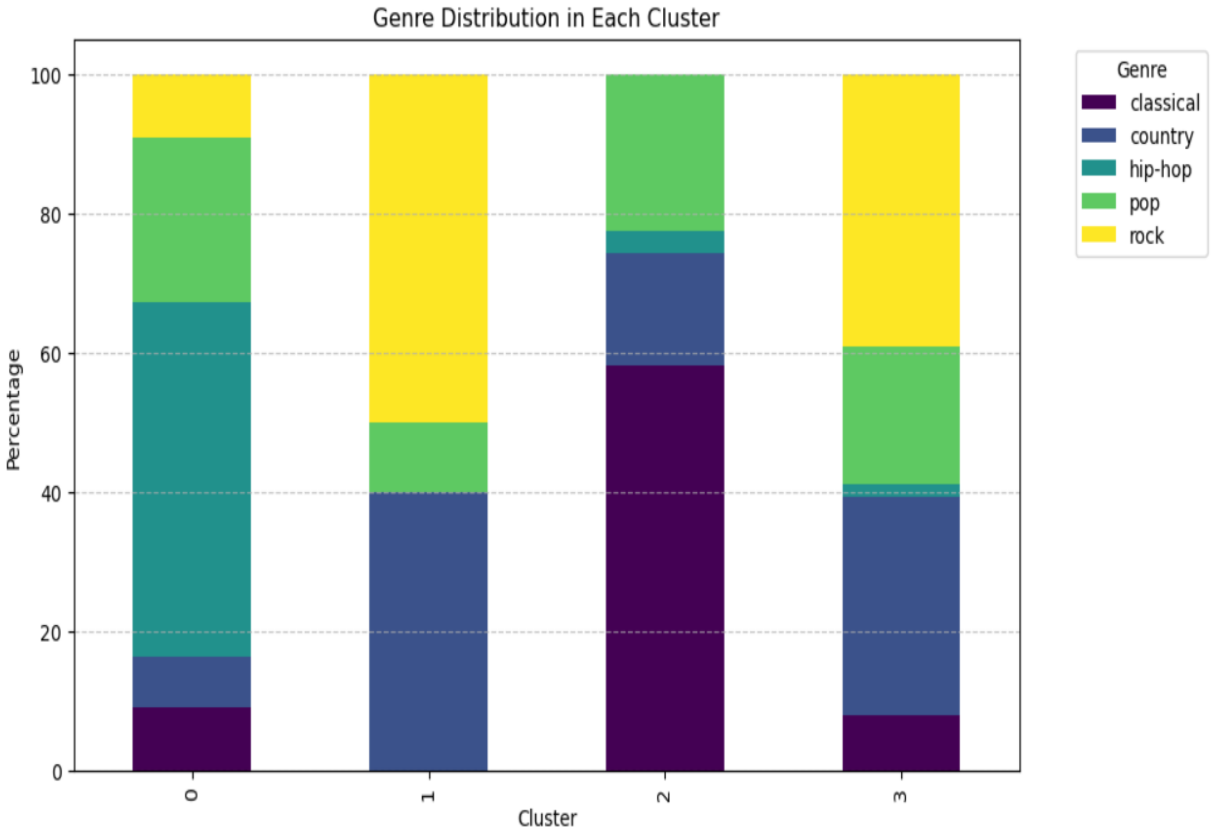
To identify natural groupings within the music data, I used the K – Means++ clustering The optimal number of clusters was chosen based on the variance per cluster(i.e the within cluster sum of squares(wcss)) and silhouette score. Each song was assigned to a cluster based on similarities in its principal component representation. This step helped in discovering potential genre groupings without prior labels.

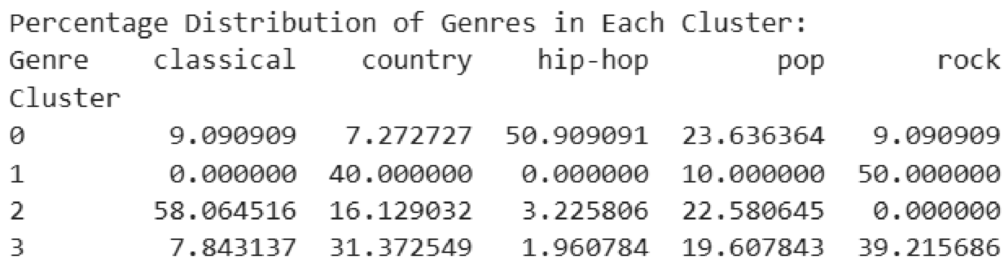
In the K-means++ algorithm, the initialisation of centroids is not done randomly but takes a probabilistic approach where probability of choosing a point as a centroid is proportional to the square of the distance from its nearest centroid. This ensures that points farther from existing centroids have a higher chance of getting selected. Thus, K-Means ++ clustering enhances the quality of clustering and also leads to faster convergence.

5. Genre Distribution Across Clusters:

Once clustering was completed , I analysed the distribution of music genres in each cluster. This helped in evaluating the effectiveness of K – Means++ in grouping similar genres together.

A bar chart was used to visualise how different genres were distributed across clusters, helping to assess whether the clustering aligned with musical intuition.





```
Percentage Distribution of Genres in Each Cluster:
Genre    classical    country    hip-hop        pop       rock
Cluster
0         9.090909   7.272727  50.909091  23.636364   9.090909
1         0.000000  40.000000   0.000000  10.000000  50.000000
2        58.064516  16.129032   3.225806  22.580645   0.000000
3         7.843137  31.372549   1.960784  19.607843  39.215686
```

Here, most of the clusters do not show a dominant genre because our clustering algorithm reflects multiple musical attributes which might be common among different music genres. For example, both ‘pop’ and ‘hip-hop’ ,use of synthesizers is common,etc.

6. Calculation of Silhouette score:

Silhouette score is an intrinsic metric used to determine the accuracy of our clustering algorithm. The value of Silhouette ranges from -1 to 1 and is given by:

$$S = \frac{b - a}{\max(a, b)}$$

It depends on two factors namely ‘Cohesion’ and ‘Adhesion’.
Cohesion:
It is the mean of the distances of the data point from all the other data points within the same cluster
Adhesion:
It is the minimum of the average distances of the data point from each of the clusters except the one that it is present in.

Silhouette score close to one implies a good and efficient clustering algorithm.

Interpretation of obtained Silhouette score:

One of the possible reasons to not achieve a high silhouette score might be the use of BoW followed by PCA. Even when BoW is the more appropriate vectorisation technique, it creates a very high dimensional and sparse vector spaces. PCA best works with dense data and thus , it might not be very efficient with the BoW vectors.

6. Genre Classification using K-Nearest Neighbours:

With the clusters formed, the next step is to predict the genre of new songs. We employed the K-Nearest neighbours(KNN) algorithm , which classifies a song based on the majority genre of its k-closest neighbours in the PCA- reduced feature space.

The dataset was split into training and test sets, and KNN was trained on the reduced features. Predictions were then made for the test set.

7. Summary:

This music genre prediction model is built using a combination of feature extraction, dimensionality reduction, clustering and classification techniques. First, Bag Of Words is used to extract meaningful textual features from the dataset, providing a structured representation of musical attributes. To enhance efficiency , PCA is applied to reduce dimensionality while preserving key information. The model then combines three different feature vectors into a single vector using weighted mean method. For initial pattern recognition, K-Means++ clustering is used to group similar songs, allowing the model to learn underlying genre structures in an unsupervised manner. Finally, K-Nearest Neighbours(KNN) is employed for classification , assigning a genre label on the closest neighbours in the transformed feature space.