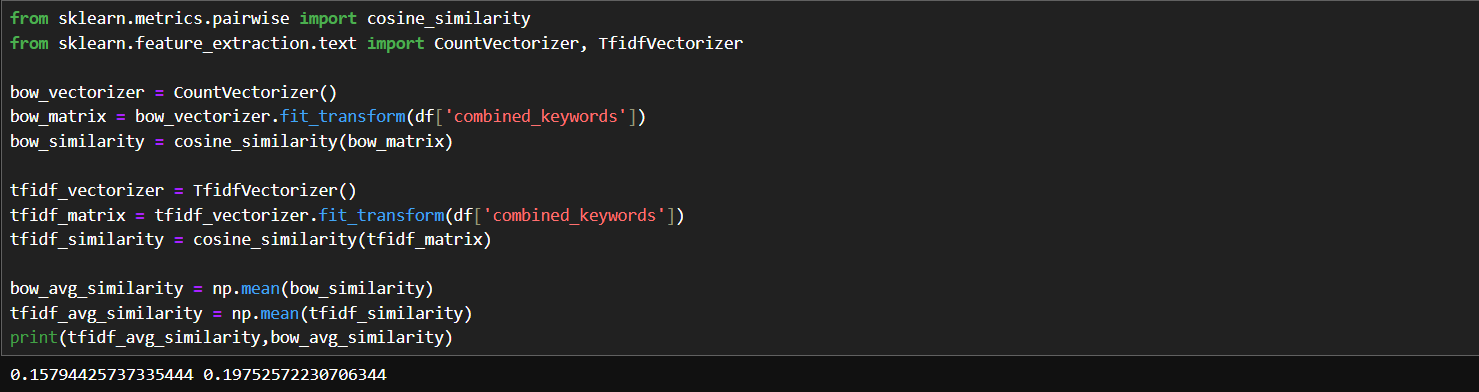
Good day,  
I am Pranav Agrawal (24CS10121).  
  
I started off this project by going through the various resources provided in the task itself. I read all the steps and algorithms which I was supposed to know and implement.

First, I start off pretty simple by importing libraries and reading the dataset, storing it in a Pandas dataframe.

Second was generating the vectors for the keywords.  
This was difficult for me as I am new to NLP things but research helped me implement it.

I have implemented Tf-Idf over Bag of Words (BoW) because it gave a better lower Tf-Idf score than Bow in vectorization of the dataset which makes us realize that it is better than Bow in distinguishing between songs and makes better distinctions and hence, leading to accurate results. 

Also, Tf-Idf reduces the impact of the words that appear frequently across all songs, allowing more unique keywords to be included in the vectorization. Tf-Idf looks for the keyword in the whole of corpus instead of only in the document.

I had some difficulty in implementing the vectorization. I also vectorized the whole “combined\_keywords” column once which gave some promising results

I implemented PCA on all three vectors which led to three (147,2) shape vectors.

I used PCA to identify the principal components which we use to differentiate between features and then use those components to transform data to a 2-dimensional vector.   
The algorithm for PCA is simple and as follows:

Standardizing or centring the data

Calculate covariance matrix and then find the eigenvalues and eigenvectors of the matrix.

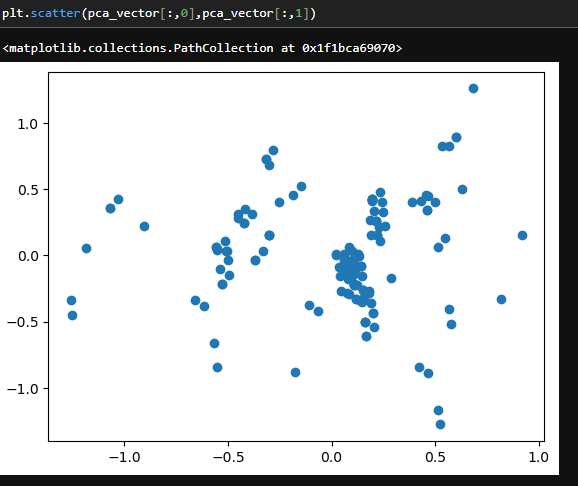
We sort the eigenvectors based on their eigenvalues from highest to lowest

We, then calculate the explained variance for each eigenvector

Then, we reduce our data to the desired number of principal components, in this case, 2

Then, we can just plot the data as the dimension is reduced.

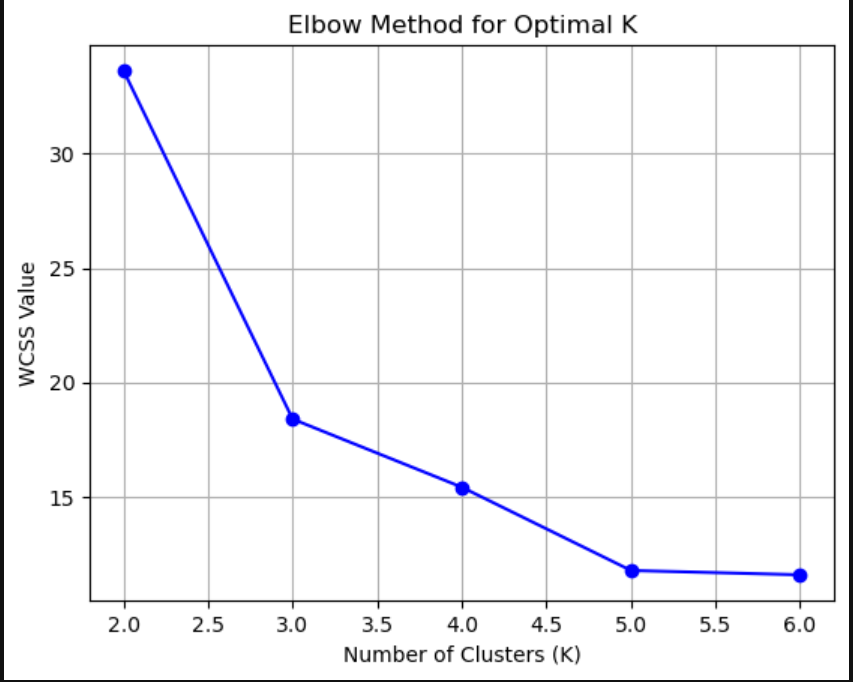
Now comes the part of merging the 3 vectors together.  
  
I simply took average of all three vectors, as all 3 vectors are equally important in the final vector. All 3 keywords (instrument, mood and style) have same weight in our analysis.

This is the plot of the final PCA vector that I got.

Now, we proceed to the clustering part of the task. For clustering, I have implemented the K-Means algorithm. I created a class called Kmeans with various methods to help us along the way. It randomly selects k points and assumes them as centroids and calculates distances from each point to them. It then assigns every point to one of the centroids based on its closeness to the point. Then, it takes the average of each cluster and calculates new centroid. Finally, it repeats the process for a fixed number of iterations.

I have also included another method to calculate WCSS (within cluster sum of squares) which is needed in the elbow method which I have discussed below.

To calculate the number of clusters(k), I have implemented the elbow method. It calculates WCSS of each k in a given range and plots them. When there is a distortion in the graph, that is our k.



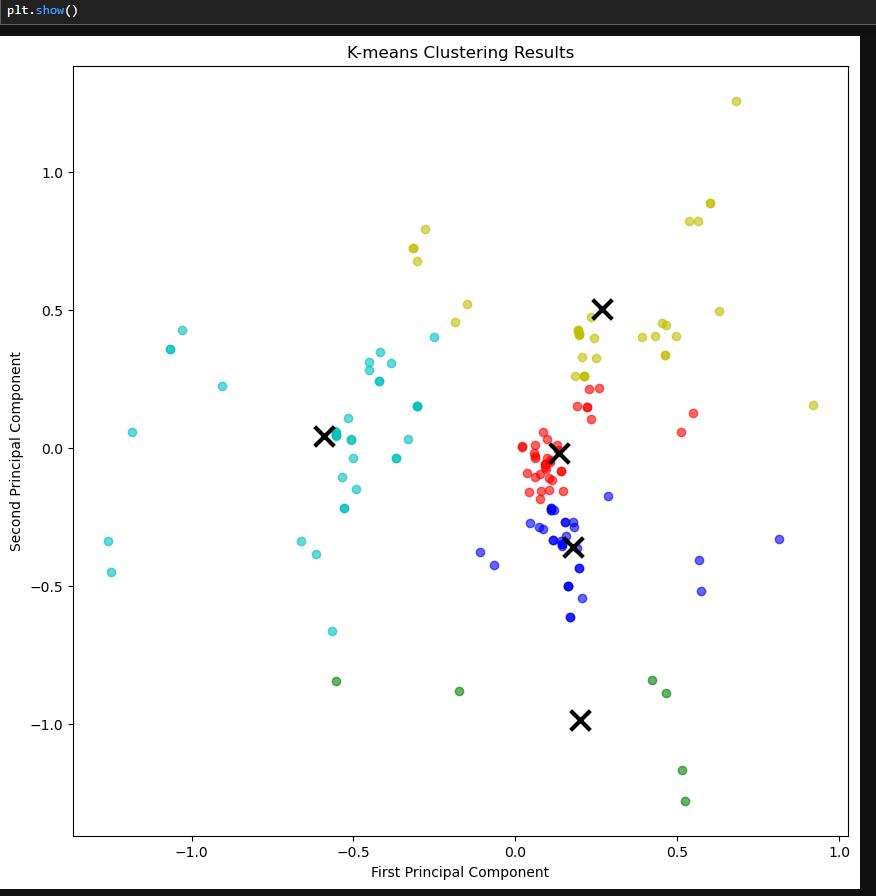
Using this plot, we can say that our k is 5.

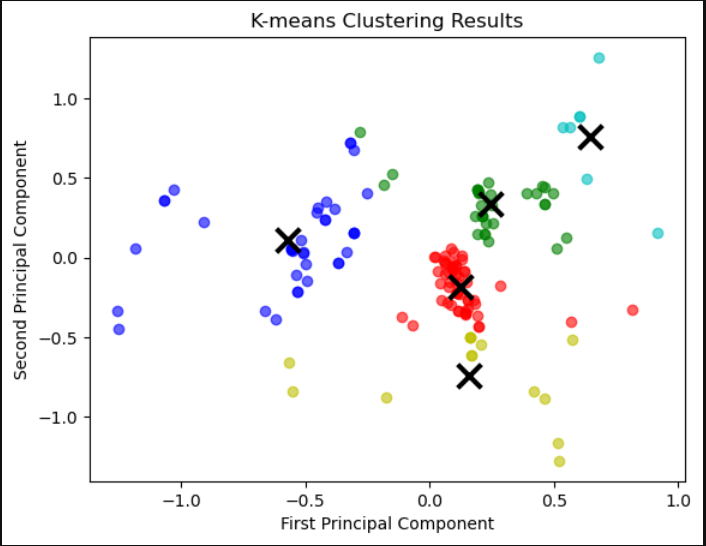
This is further backed by the fact that we have 5 genres in the dataset. Hence, they will create 5 clusters.

So, k is 5.

Taking k as 5, I plotted the clustered graph with different cluster as different colours.

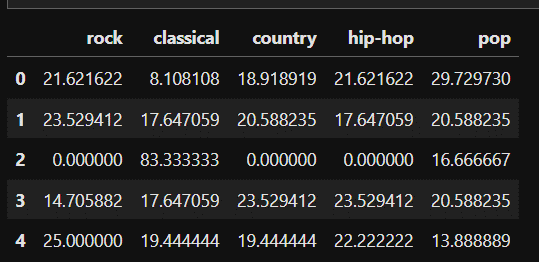
Here are the results.   
The graph will change on each run due to the nature of Kmeans algorithm (initialization of k is random).





Now, we move on to the analysis part

First, the percentage of ground truth in each cluster -



This shows that:

Cluster-0 is mainly pop with some rock and hip-hop

Cluster-1 is a perfect blend of all genres

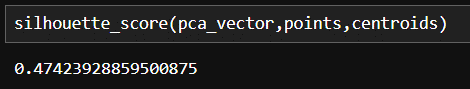
Cluster-2 is majorly classical (80%) with a little bit of pop

Cluster-3 is a fair share of country and hip-hop

Cluster-4 is mainly rock with some mixed genres

Some of the clusters match with their true genres but overall, it’s a very diverse and different from our expectations.

Now, we calculate the silhouette score of our analysis -



Our Silhouette score is 0.47, which shows moderate level of cluster separation. This indicates some overlap between clusters, which prompts us that k=5 is probably not a good choice.

For the prediction part, I have an idea as to what to do but I couldn’t implement it.

I thought of appending the new keywords into the Dataframe and then doing the whole

Tf-Idf, PCA and clustering again to get accurate results. But I was not successful in doing so.