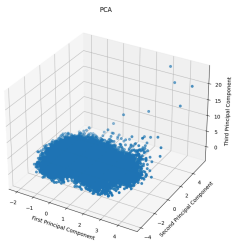
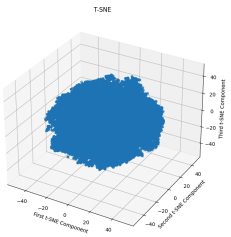
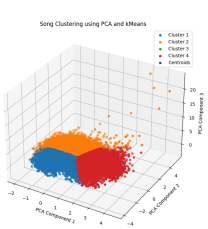
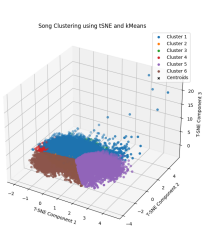
For data cleaning, I first examined the outcome variable, which is the 10 classes of music genre. I found out that there are 5 rows of missing data within that column, so I dropped them so that we have 50,000 rows of data evenly split into the 10 different genres, with 5,000 rows for each. I then converted the music genre column into 10 columns of 0s and 1s and used that as the output variable for later uses. After this, I examined other feature variables one by one to see if there were any missing values and if they were normally distributed. I make sure that they have no missing values. There are missing values in the column duration and tempo. I replaced them with the median value within their genre to ensure accuracy and fairness across different genres. That's why I didn't use the overall mean or median. I converted the obtained date column into numerical values because I think they have a numeric meaning, with higher values indicating later dates, which are newer. For other categorical variables (key and mode), I made them into dummy variables like how I dealt with the outcome variable music genre. For the quantitative variables, some are normally distributed and some are not. For normally distributed ones, I standardized them to ensure the same scale and better results. I didn't standardize any of the categorical variables though, for the purpose of doing dimensional reduction later. For other quantitative variables that don't seem to follow a normal distribution, I applied power transform to them before standardizing them, using the box cox method for columns where all values are positive (no 0s) and the yeo johnson method for columns that contain 0s. For extra credit, it is interesting and surprising to see that there are almost twice as many songs written in a major than songs written in a minor. Intuitively, I thought they would be of an equal or similar ratio because major songs are usually upbeat and in a happier mood whereas minor songs are usually melancholic and in a down mood. This shows that most songs on Spotify are happy, which is contrary to my personal experience. I don't have a preference for happy songs or sad songs, but my Spotify playlist consists of roughly same number of happy and sad songs. It's also surprising to see that energy is not normally distributed. Most songs have a higher energy, which corresponds to the major vs. minor pattern.

Before doing anything else, I did the train test split to evaluate the performance of a model on unseen data. I set the test size to 0.1 and used the stratify method using the music genre column to ensure that both the train set and the test set contain the same ratio of different genres. This way, they correspond to the same equal distribution in the original dataset. The train test contains 45,000 songs, 4500 songs from each genre. The test set contains 5,000 songs, 500 songs from each genre. For dimensionality reduction, I did PCA and t-SNE respectively. Before doing them, I first calculated the covariance matrix and the eigenvalues associated with each feature. It seems like the first three components are greater than 1: 2.82902510e+00 1.70038784e+00 1.10278320e+00. Therefore, I chose three components for PCA and t-SNE.









As we can see, dimensionality reduction methods don't work as well as expected for this dataset. Both PCA and t-SNE generate vague clusterings that seem like one huge cluster to me. PCA has a few outliers. Nevertheless, I continue to do clustering methods building on this PCA and t-SNE. I used the Silhouette method to determine the optimal number of clusters and then used kMeans with that number (k) to produce a plot that represents each song as a dot in a 3D space in the color of its cluster. I also drew the elbow curve to indicate this process of finding the optimal number of clusters. It turns out that the optimal number of clusters is 4 for PCA and 6 for t-SNE. However, the 3D space embedding for PCA shows only 3 clusters (we can only see points from Cluster 1, 2, 4 but barely see any points from Cluster 3) and the 3D space embedding for t-SNE shows only 4 clusters (we can barrel see any points from Cluster 2 and 3). This shows the clustering methods didn't do clustering well. Furthermore, since we already have the 10 labels, we know the correct number of clusterings should be 10, but neither methods captured that.

Finally, I trained 3 linear SVMs using the original input data without any dimensionality reduction, the input data after PCA with three components, and the input data after t-SNE with three components, respectively, to compare the performance of them with each other. I computed the ROC curve, AUROC, precision-recall curve, and AUPRC for each class and the micro-average over all classes. Since the roc_auc_score and precision_recall_curve functions are designed for binary classification tasks, I didn't use them to calculate the AUROC and AUPRC. Instead, I binarized the output, calculated these metrics for each class, and then averaged them since this is a multiclass classification problem. As a result, I got the highest AUROC from the SVM trained with the original input data without any dimensionality: 0.91. The SVM trained with PCA data has a slightly lower AUROC of 0.82 and the SVM trained with t-SNE data has the lowest AUROC of 0.76.