MUSIC GENRE CLASSIFICATION

Mentee: Yitong Shan
Mentor: Alex Ziyu Jiang

Motivation 

Inspired by the sheer amount of music genres in Spotify and Apple Music, we came up with the idea to process audio data and used machine learning techniques to classify songs into different genres.

Classification method: K-Nearest Neighbors

Model validation method: K-Fold cross validation

(we'll explain them next :))

Data Processing

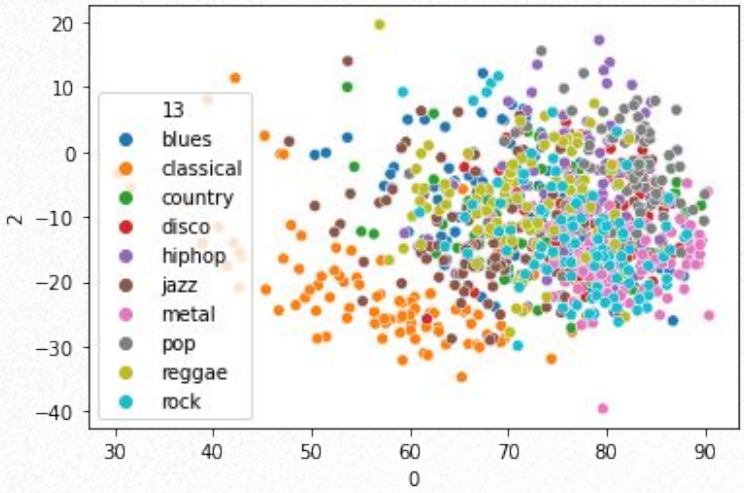
Extract features: divide each song into several sections, extract 13 features from each section using Mel Frequency Cepstral Coefficient (MFCC) method, and calculate the mean and covariance of the features. The results represents the features of the song.

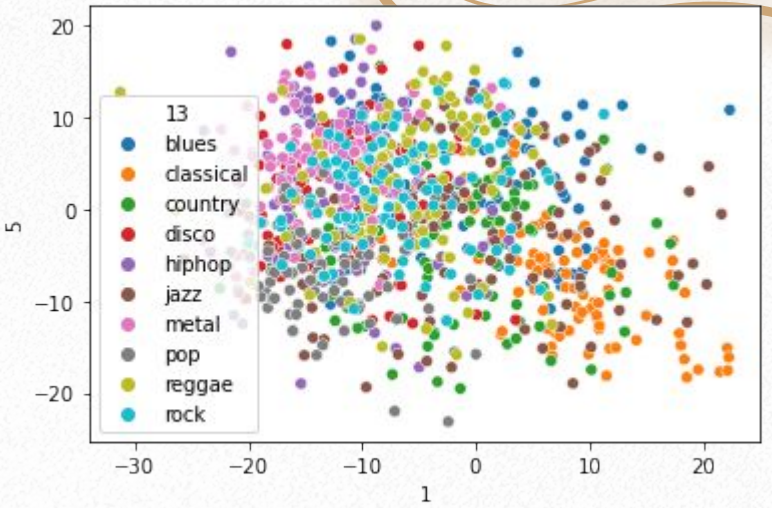
Estimate distributions: Estimate the mean and covariance for each song using the extracted features based on Normal assumption.

Modify KL divergence based on the estimated mean and covariance, the resulted formula is used to calculate the distance between two songs (distributions)

$$D_{\mathrm{KL}}\left(\mathcal{N}_0 \parallel \mathcal{N}_1\right) = \frac{1}{2}\left(\operatorname{tr}\left(\Sigma_1^{-1}\Sigma_0\right) + \left(\mu_1 - \mu_0\right)^{\mathsf{T}}\Sigma_1^{-1}\left(\mu_1 - \mu_0\right) - k + \ln\left(\frac{\det\Sigma_1}{\det\Sigma_0}\right)\right)$$

Exploratory Data Analysis





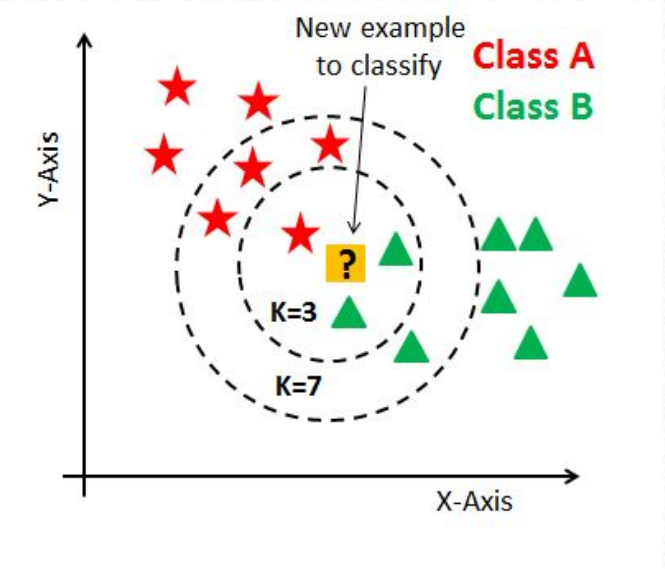
The distribution of songs with features as the parameters: feature 0 and 1 clustered songs with the highest distinctions. Which means they are the key features to distinguish genres.The orange dots are far apart from other dots, which suggests that the classical music could be very different from other music genres.

Methods

K-Nearest neighbors: KNN makes predictions based on the K nearest instances in the training data.

To find the neighbors, we need to find the distance between the training data and other data points.

By choosing different values of K, we might get different predictions; Our goal is to find the K value with **the highest prediction accuracy**.

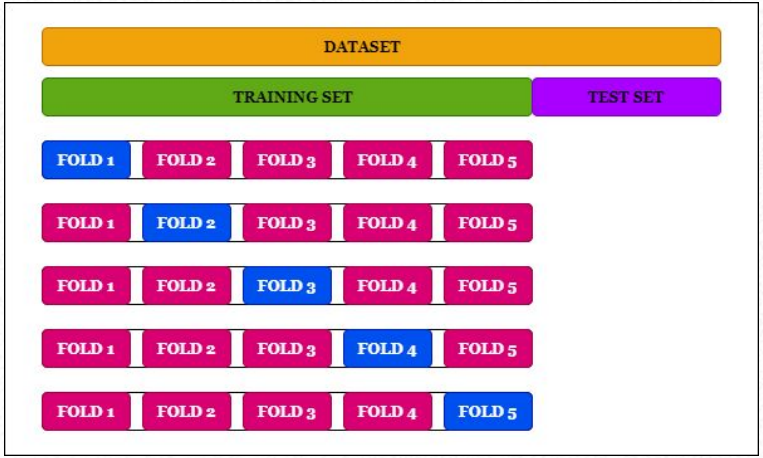


Methods

K-fold Cross Validation:

Method: divide the **dataset**, hold out 20% of the data as the **test set**, and the rest 80% are **training sets**. Each time, one of the k subsets is hold out as the **validation set**, and the other k-1 subsets are put together to form a **training set.** The KNN classification is repeated k times on the **training set.**

Advantage: it matters less how the data gets divided, stabilizes our result.

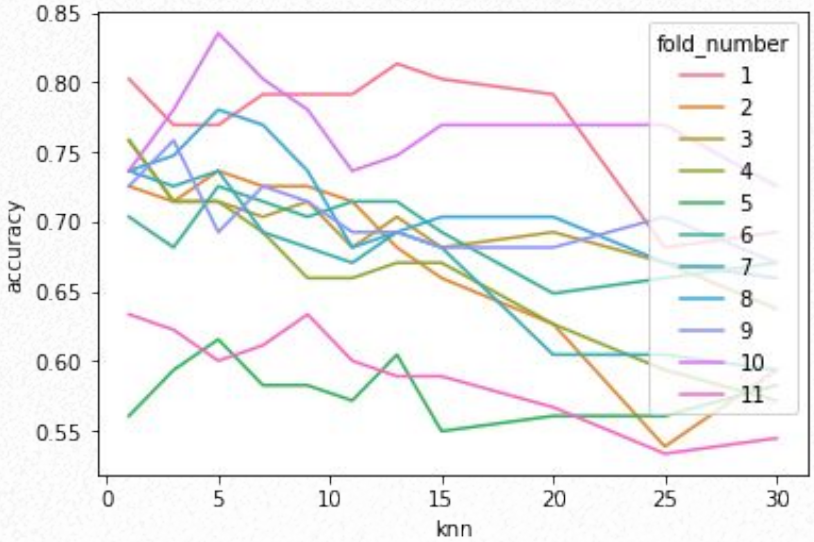


Methods

```
dat_kfold2 = pd.DataFrame()
n=1
np.random.shuffle(whole2)
leng = len(testSet)

for train_index, test_index in kf.split(whole2):
    W_train, W_test = whole2[train_index], whole2[test_index]
    predictions = []
    for i in [1,3,5,7,9,11,13,15,20,25,30]:
        for x in range (leng):
            predictions.append(nearestClass(getNeighbors(W_train, testSet[x] , i)))
        accuracy2 = getAccuracy(testSet , predictions)
        dat_kfold2 = dat_kfold2.append({'fold_number': str(n),'knn': i,'accuracy':accuracy2},
ignore_index=True)
        predictions = []
    n+=1
```

Results



	knn	accuracy
2	5.0	0.719880
0	1.0	0.715917
1	3.0	0.710911
3	7.0	0.709901
4	9.0	0.701931
6	13.0	0.690898
5	11.0	0.682917
7	15.0	0.679909
8	20.0	0.660906
9	25.0	0.634898
10	30.0	0.630914

By averaging the accuracies on the validation set over each fold for each k value, we get the result that when k=5, the accuracy is the highest. Indicates when we choose 5 neighbors to classify a specific data, the prediction is closest to its actual class.

Exploration

We use k=5 to get the prediction accuracies of each genre.

```
leng = len(testSet)
dat = pd.DataFrame()

def generatelist(n, testset_genre, predictions_genre):
    global dat
    for x in range (leng):
        genre = testSet[x][2]
        if genre == n:
            testset_genre.append(testSet[x])
            predictions_genre.append(nearestClass(getNeighbors(trainingSet
,testSet[x] , 5)))
    accuracy = getAccuracy(testset_genre , predictions_genre)
    dat = dat.append({'genre': results[n],'knn': 5,'accuracy':accuracy},
ignore_index=True)
    predictions_genre = []
```

Exploration

	genre	knn	accuracy
1	classical	5.0	0.944444
7	pop	5.0	0.900000
4	hiphop	5.0	0.866667
2	country	5.0	0.750000
5	jazz	5.0	0.714286
6	metal	5.0	0.705882
8	reggae	5.0	0.687500
3	disco	5.0	0.631579
0	blues	5.0	0.578947
9	rock	5.0	0.461538

Conclusion: by using k=5, we see ‘classical’ genre gives the highest accuracy, and ‘rock’ genre has the lowest prediction accuracy.

Results

- Using k-fold methods, we generated the result that k=5 gives us the highest prediction accuracy under the KNN model.
- The prediction accuracy of the dataset is about 72%
- Among the 10 genres, classical genre has the highest accuracy, and rock genre has the lowest.

Reference

https://www.cs.cmu.edu/~schneide/tut5/node42.html
https://scikit-learn.org/stable/modules/cross_validation.html
https://data-flair.training/blogs/python-project-music-genre-classification/
James, Gareth, et al. *An Introduction to Statistical Learning: With Applications in R*.

THANK YOU