Capstone Project Introduction to Machine Learning
Classification flavor

Preamble: Spotify released an API that reports the audio features of its songs, such as "tempo" or "energy". Here, we will use these features of 50k randomly picked songs to predict the genre that the song belongs to.

Scope: This is a "for real" project, that includes parts of all aspects of the class. It is designed to stretch your abilities. No one is expecting perfection – the question is what you can come up with in a couple of weeks, while you have other ongoing commitments, which simulates typical industry conditions. If achieving reasonable performance, you will be able to use this capstone project as a "portfolio" item that you can highlight in interviews for internships and jobs. Note that this is not a toy dataset or project. As real data always has (many) real issues, any success in classification will be hard earned and not easily forthcoming.

Academic integrity: You are expected to do this project by yourself, individually, so that we are able to determine *your* ability to solve machine learning problems. We'll be on the lookout for suspicious similarities, so please refrain from copying off someone else (also note that the idiosyncratic seed keys the correct answers to you, and not anyone else, making it effectively pointless to copy off others).

Data: The data is contained in the file musicData.csv. In this file, the first row represents the column headers. Each row after that represents data from one song.

The columns represent, in order (from left to right):
Column 1: unique Spotify ID of each song
Column 2: artist name
Column 3: song name
Column 4: popularity of the music (what percentage of users know about it, from 0 to 99%)
Column 5: acousticness (an audio feature) on a scale from 0 to 1
Column 6: danceability (an audio feature) on a scale from 0 to 1
Column 7: the duration of the music (in milliseconds)
Column 8: energy (an auditory feature) on a scale from 0 to 1
Column 9: instrumentality (an audio feature) on a scale from 0 to 1
Column 10: key of the song (in musical notation)
Column 11: liveness (an audio feature) on a scale from 0 to 1
Column 12: loudness (in dB relative to some threshold)
Column 13: mode of the song (musical)
Column 14: speechiness (an audio feature), on a scale from 0 to 1
Column 15: tempo (in beats)
Column 16: obtained date (when was this information obtained from Spotify)
Column 17: valence (an audio feature), on a scale from 0 to 1
Column 18: Genre of the song (there are 10 different genres, e.g. "Rock" or "Country")

Here is what we want you to do:
*Download the file musicData.csv from Brightspace
*Write code that creates a classification model (of your choice, can be anything we covered in class – including SVM, trees, forests, boosting methods, neural networks, etc.; for good classification performance, make sure to include a suitable dimensionality reduction and clustering step before attempting the classification) that predicts the genre of a song in the test set from the rest of the data.
*In the code, before anything else happens, we want you to initialize the seed of the random number generator with your NYU N-number. For instance, if your N-number (on the back of your NYU ID) is N18994097, we would expect the first lines of your code to be:
import random
random.seed(18994097)
Note: Don't use "18994097", unless that is your N-number. Use yours instead. This is so that the correct answers (which depend on which movie ratings go into the test set) are specific to *you*.
*Make sure to do the following train/test split: For *each* genre, use 500 randomly picked songs for the test set and the other 4500 songs from that genre for the training set. So the complete test set will be 5000x1 randomly picked genres (one per song, 500 from each genre). Use all the other data in the training set and make sure there is no leakage.
*Use this test set to determine the AUC of your model. Try to get the AUC as high as possible without having any leakage between training and test set.
*Write a brief report (1-2 pages) as to how you built your model, how you made your design choices (why you did what you did) and addressing how you handled the challenges below. Make sure to state your final AUC at the bottom of the report and please include a plot of the ROC curve in your report. As dimensionality reduction will be critical, please also include a visualization of the genres as clusters in the lower dimensional space (lower than the dimensionality of the original data) you uncovered and include a comment as to what you think about this space/clustering. Also make sure to comment on what you think is the most important factor that underlies your classification success.
*For extra credit, include some interesting non-trivial observation or data visualization in your report.
*Upload both the report (pdf only) and your code (some kind of python or notebook format only) to the Brightspace portal by the due date.

Some key challenges you can expect when taking on this project:
*There is randomly missing data, e.g. some of the durations of some of the songs are missing, as well as some of the auditory feature values. There are not many missing values, but you have to handle them somehow, either by imputation or by removing the missing data in some reasonable way.
*The acoustic features are unlikely to be normally distributed.
*Some of the data is provided in string format, e.g. the key. This will need to be transformed into numerical data to be useful.
*Some of the data is provided in categorical format, e.g. mode. This will need to be dummy coded.
*The category labels of the genres will need to be transformed into numerical labels.
*As this is a multi-class classification (not just binary), you need to be careful – only if the predicted classification of a genre in the test set matches the actual genre of a song is the classification correct. In other words, there are many more ways to be wrong than to be right.
*Make sure not to normalize categorical values (like mode) for the purposes of doing dimensionality reduction.
*There might be information in the linguistic properties of artist and song, but you don't have to include that you in your model.