Introduction to Data Science – DS GA 1001
Capstone project

The purpose of this capstone project is to tie everything we learned in this class together. This might be challenging in the short term, but is consistently rated by students as being extremely valuable and useful in the long run. The cover story this time is that you are working as a Data Scientist for Spotify. You have data on a set of 52,000 songs and you want to better understand what makes music popular as well as the audio features that make up specific genres. Historically, this domain was dominated by musicians and music theorists, but is increasingly the domain of the data scientist.
This is where you come in: Can you provide the value that justifies your rather high salary?

Mission command preamble: As usual, we won't tell you *how* to do something. That is up to you and allows you to showcase your creative problem-solving skills. However, we will pose the questions that you should answer by interrogating the data.

Format: The project consist of your answers to 10 (equally-weighed, grade-wise) questions. Each answer *must* include some **text** (describing both what you *did* and what you *found,* i.e. explicitly stating the answer to the question), a **figure** that illustrates the findings and some **numbers** (e.g. test statistics, confidence intervals, p-values or the like). Please save it as a pdf document. This document should be 5-7 pages long (arbitrary font size and margins). About ½ a page/question is reasonable. In addition, open your document with a title page where you introduce your group (and group name), state author contributions as well as statements as to how you handled preprocessing (e.g. dimension reduction, data cleaning and data transformations), as this will apply to all answers.

Academic integrity: You are expected to do this project as a group. So make sure this works reflects your intellectual contribution – not that of third parties. Feel free to use generative AI like chatGPT to aid you in this task, but make sure to specify in the author contributions how you used if, if you did. There are enough degrees of freedom (e.g. how to clean the data, what variables to compare, aesthetic choices in the figures, etc.) that no two reports will be alike. We'll be on the lookout for suspicious similarities, so please refrain from collaborating.
To prevent cheating (please don't do this – it is easily detected), it is very important that you – at the beginning of the code file – seed the random number generator with the N-number of one of your team members (specify which one). That way, the correct answers will be keyed to your own solution (as this matters, e.g. for the specific train/test split or bootstrapping).
As N-numbers are unique, this will also protect your work from plagiarism.
Failure to seed the RNG in this way will also result in the loss of grade points.

Deliverables: Upload two files to the Brightspace portal by the due date in the sittyba:
*A pdf (the "project report") that contains your answers to the questions, as well as an introductory paragraph about preprocessing, how you seeded the RNG, etc.
*A .py file with the code that performed the data analysis and created the figures. This will help us understand what you were trying to attempt – in particular if you get the "wrong" answer.

We do wish you all the best in executing on these instructions. We aimed at an optimal balance between specificity and implementation leeway, while still allowing us to grade the projects in a **fast**, **fair** and **faithful** (=consistent and accurate) manner (FFF).
Everything we ask for should be doable from what was covered in this course.

Description of dataset: This dataset consists of data on 52,000 songs that were randomly picked from a variety of genres sorted in alphabetic order (a as in "acoustic" to h as in "hiphop"). For the purposes of this analysis, you can assume that the data for one song are independent for data from other songs.

This data is stored in the file "spotify52kData.csv", as follows:
Row 1: Column headers
Row 2-52001: Specific individual songs
Column 1: **songNumber** – the track ID of the song, from 0 to 51999.
Column 2: **artist(s)** – the artist(s) who are credited with creating the song.
Column 3: **album_name** – the name of the album
Column 4: **track_name** – the title of the specific track corresponding to the track ID
Column 5: **popularity** – this is an important metric provided by spotify, an integer from 0 to 100, where a higher number corresponds to a higher number of plays on spotify.
Column 6: **duration** – this is the duration of the song in ms. A ms is a millisecond. There are a thousand milliseconds in a second and 60 seconds in a minute.
Column 7: **explicit** – this is a binary (Boolean) categorical variable. If it is true, the lyrics of the track contain explicit language, e.g. foul language, swear words or content that some consider indecent.
Column 8: **danceability** – this is an audio feature provided by the Spotify API. It tries to quantify how easy it is to dance to the song (presumably capturing tempo and beat), and varies from 0 to 1.
Column 9: **energy** - this is an audio feature provided by the Spotify API. It tries to quantify how "hard" a song goes. Intense songs have more energy, softer/melodic songs lower energy, it varies from 0 to 1.
Column 10: **key** – what is the key of the song, from A to G# (mapped to categories 0 to 11).
Column 11: **loudness** – average loudness of a track in dB (decibels)
Column 12: **mode** – this is a binary categorical variable. 1 = song is in major, 0 – song is in minor
Column 13: **speechiness** – quantifies how much of the song is spoken, varying from 0 (fully instrumental songs) to 1 (songs that consist entirely of spoken words).
Column 14: **acousticness** – varies from 0 (song contains exclusively synthesized sounds) to 1 (song features exclusively acoustic instruments like acoustic guitars, pianos or orchestral instruments).
Column 15: **instrumentalness** – basically the inverse of speechiness, varying from 1 (for songs without any vocals) to 0.
Column 16: **liveness** - this is an audio feature provided by the Spotify API. It tries to quantify how likely the recording was live in front of an audience (values close to 1) vs. how likely it was recorded in a studio without a live audience (values close to 0).
Column 17: **valence** - this is an audio feature provided by the Spotify API. It tries to quantify how uplifting a song is. Songs with a positive mood =close to 1 and songs with a negative mood =close to 0
Column 18: **tempo** – speed of the song in beats per minute (BPM)
Column 19: **time_signature** – how many beats there are in a measure (usually 4 or 3)
Column 20: **track_genre** – genre assigned by spotify, e.g. "blues" or "classical". 1k songs per genre.

In addition, there is a file ("starRatings.csv") that contains explicit feedback, specifically star ratings from 10k users on 5k songs they listened to, on a scale from 0 (lowest) to 4 (highest). In this file, there are no headers. Each row corresponds to a user and each column to a song, specifically to the first 5k rows (songs) in the spotify52kData.csv dataset, in the same order. Missing data is represented as nans.

Note that we did most of the data munging and coding for you already but you still need to handle missing data in some way (e.g. by row-wise removal, element-wise removal, imputation, masking, etc.). Also, if there are skewed distributions, extreme values might also have to be handled.

Corporate needs you to find the answers to these questions:

1) Is there a relationship between song length and popularity of a song? If so, is it positive or negative?
2) Are explicitly rated songs more popular than songs that are not explicit?
3) Are songs in major key more popular than songs in minor key?
4) Which of the following 10 song features: *duration*, *danceability*, *energy*, *loudness*, *speechiness*, *acousticness*, *instrumentalness*, *liveness*, *valence* and *tempo* predicts popularity best? How good is this model?
5) Building a model that uses *all* of the song features mentioned in question 1, how well can you predict popularity? How much (if at all) is this model improved compared to the model in question 4). How do you account for this? What happens if you regularize your model?
6) When considering the 10 song features in the previous question, how many meaningful principal components can you extract? What proportion of the variance do these principal components account for? Using these principal components, how many clusters can you identify? Do these clusters reasonably correspond to the genre labels in column 20 of the data?
7) Can you predict whether a song is in major or minor key from *valence* using logistic regression or a support vector machine? If so, how good is this prediction? If not, is there a better one?
8) Can you predict genre by using the 10 song features from question 4 directly or the principal components you extracted in question 6 with a neural network? How well does this work?
9) In recommender systems, the popularity based model is an important baseline. We have a two part question in this regard: a) Is there a relationship between popularity and average star rating for the 5k songs we have explicit feedback for? b) Which 10 songs are in the "greatest hits" (out of the 5k songs), on the basis of the popularity based model?
10) You want to create a "personal mixtape" for all 10k users we have explicit feedback for. This mixtape contains individualized recommendations as to which 10 songs (out of the 5k) a given user will enjoy most. How do these recommendations compare to the "greatest hits" from the previous question and how good is your recommender system in making recommendations?

Extra credit: Tell us something interesting about this dataset that is not trivial and not already part of an answer (implied or explicitly) to these enumerated questions [Suggestion: Do something with the number of beats per measure, something with the key, or something with the song or album titles]

Hints:

*Beware of off-by-one errors. This document and the csv data files index from 1, but Python indexes from 0. Make sure to keep track of this.
*In order to answer some of these questions, you might have to apply a dimension reduction method first. Similarly, you might have to reduce variables to their summary statistics.
*In order to do some analyses, you will have to clean the data first, either by removing or imputing missing data or handling it in some other way (either is fine, but explain and justify what you did)
*If you encounter skewed data, you might want to transform the data first, e.g. by z-scoring
*To clarify: When talking about "principal components" above, we mean the transformed data, rotated into the new coordinate system by the PCA.
*Avoid overfitting with cross-validation methods when making predictions.
*How well your model predicts can be assessed with RMSE or R^2 for regression models, and AUC for classification models. Use a suitable metric such as average precision for recommender systems.
*You can use conventional choices of alpha (e.g. 0.05) or confidence intervals (e.g. 95%) throughout.
*Make sure to actually answer all questions (particularly for multi-part of questions) that were asked explicitly, for full credit. Make sure to follow style and logistics instructions (e.g. no seeding of RNG with N-number, no statement of author contributions, etc.) to avoid losing points.