Aprendizagem Automática (APRAU)

Mestrado em Engenharia Informática

Assignment

---

1 Objectives

The **general objective** of this assignment is to apply the different machine learning methods on a dataset, analyze and understand the results obtained.

2 Exploratory Data Analysis (EDA)

The dataset you will be working with consists of detailed metadata and audio analyses for a wide collection of music tracks across various genres. It includes track-level information such as popularity, tempo, energy, movement, and other measurable characteristics that will be used for analysis such as regression, classification and feature selection. Each instance in the dataset represents a single record with two associated target variables: a categorical label indicating the item's class (used for classification tasks) and a numerical score representing its popularity or impact (used for regression tasks).

Features
The dataset contains the following types of features:

- duration_1 ... duration_5: Different measurements or transformations of the item's duration;
- loudness_level: A classification of the track's overall loudness;
- popularity_level: Discretized level of popularity;
- tempo_class: A classification of the song's tempo (beats per minute);
- time_signature: Represents how many beats are in each bar of music;
- key_mode: A standardized numerical representation combining the song's key and mode;
- artist_song_count: A standardized value representing the total number of songs by the track's artist in the dataset;
- album_freq: A standardized value likely representing how frequently the album (that this track belongs to) appears in the dataset;
- movement_index: A derived audio feature that measures the amount of change in the rhythmic pattern throughout the track;
- intensity_level: A measure of the perceptual intensity or power of a song;
- verbal_density: Represents the density of vocals or lyrics in the track;
- purity_score: Measure the degree of acoustic purity;
- positivity_index: A measure of the track's positive or happy emotional valence;
- activity_rate: Describes the level of activity or energy in the track;
- loudness_intensity: Derived combination of loudness and intensity;

- happy_dance: Index relating positive mood and movement;
- acoustics_instrumental: Combined acousticness and instrumentalness;
- artists_avg_popularity: Average popularity score of the creator(s);
- tempo_vs_genre: Relation between tempo and category;
- energy_rank_pct: The percentile rank of the track's energy level compared to other tracks in the dataset;
- loud_energy_ratio: Ratio between loudness and energy-related properties;
- mood_pca: Mood component derived from PCA;
- mood_cluster: Mood grouping obtained from clustering;
- acoustic_valence_mood_cluster: Cluster combining acoustic and valence features;
- explicit: Binary flag indicating explicit language;
- signal_strength: Strength or energy of the signal;
- mode_indicator: Binary indicator of mode (major/minor type);
- focus_factor: Degree of instrumental presence;
- ambient_level: Measure of background or live quality;
- key_sin and key_cos: Circular encoding of harmonic key;
- duration_log: Logarithmic transformation of the duration;
- duration_log_z: Standardized (z-score) version of the log-duration;
- time_signature_class_boolean: Simplified binary version of time signature;
- loudness_yeo: Yeo–Johnson transformation of loudness value;
- is_instrumental: Binary indicator of instrumental items;
- is_dance_hit: Binary indicator of dance-oriented items;
- temp_zscore: Standardized (z-score) version of tempo;
- resonance_factor: Measure of the prominence of resonant frequencies;
- timbre_index: A abstract representation of the timbral quality of the track;
- echo_constant: Measure of the presence and intensity of echo;
- distorted_movement: Measure fluctuations in movement patterns;
- signal_power: Measure of the power of the audio signal;
- target_class: Categorical class label (class_1, class_2, . . . , Class_N);
- target_regression: Continuous success/impact score.

What will be made available for you to use in building your models is not be the original dataset, but rather a subset of it. The features will be provided in csv files, where each line (of the csv file) corresponds to one music track.

Each group will receive a dataset containing three different music classes, and **each group must work with a different set of classes**. The number of different samples in the music classes vary, which means that most groups will work with an imbalanced dataset.

After assigning the data to each group, and before starting to use it, carry out an exploratory analysis to obtain more information from the dataset:

- Descriptive Statistics
- Univariate Analysis (Distribution of individual features)
- Bivariate Analysis (Correlation between features and the different target variables)

What relevant information can you extract from the Univariate and Bivariate Analysis?

3 Methods Application - Regression

Using the provided dataset, build two regression models to predict the target_regression. You may use the hold-out method to evaluate the models.

- **Simple Linear Regression**
  - Fit a model using a single feature.
  - Test different features, evaluate their performance, and select the most suitable one.
- **Multiple Linear Regression**
  - Fit a model using several audio features.
  - Experiment with different combinations of features and determine the best-performing group.
- **Comparison and Discussion**
  - Evaluate both models using appropriate metrics (e.g. R^2, MAE, RMSE).
  - Compare their performance and provide a short discussion of the results.

4 Methods Application - Classification

Consider using the following methods: Logistic Regression, Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). Applying the methods to the chosen data, try to decide which method is most appropriate for the problem, giving reasons for your choice. Use the following resampling methods for the various suggested models:

- Holdout
- Cross Validation (with $k = 5$ and $k = 10$)
- Leave One Out Cross Validation (LOOCV)
- Bootstrap

Use the evaluation metrics that you find most appropriate to evaluate the results obtained in each experiment. Analyzing the results obtained, indicate how the variance is affected by the resampling methods used.

5 Feature Selection

Can classification models obtain better results if they use just a few features instead of all available features? Evaluate this hypothesis, using regularization methods.

Note: For the following tasks, you must use the most relevant predictors, based on the results obtained in Task 5 - Feature Selection.

6 Generalized Additive Models (GAM)

Use Generalized Additive Models (GAM) to perform binary classification of your dataset. To do this, you must try to build a model that allows you, among the three classes under analysis, to identify one of them. You should test the three hypotheses and present only the one with the best results. To validate the performance of the models, use cross-validation. Evaluate the results using the evaluation metrics that you consider appropriate.

7 Decision Trees (DT) and Random Forest (RF)

1. Decision Trees (DT)
   (a) Using Decision Trees, build a classification model that allows you to differentiate the classes under analysis (without hyperparameter optimization).
   (b) Tune the Decision Tree hyperparameters, ensuring that your model is not overfitting the training data.
2. Random Forest (RF)
   (a) Using Random Forest, build a classification model that allows you to differentiate the classes under analysis (without hyperparameter optimization).
   (b) Tune the Random Forest hyperparameters, ensuring that your model is not overfitting the training data.
   (c) After building your Random Forest model, present an ordered list, with the importance of the features used by the model.
   (d) Try to correlate the results obtained in the previous question, with the Univariate and Bivariate analysis carried out in Task 2, and with the results obtained after applying the Ridge and Lasso methods in Task 5.

8 Support Vector Machine (SVM)

Using Support Vector Machines (SVMs), build a classification model that allows you to differentiate the classes under analysis. In this task you must:

- Test all possible kernels (without hyperparameter optimization);
- Tune the SVM hyperparameters, ensuring that your model is not overfitting the training data;
- Present the SVM model with the best performance on your data, justifying the choice (you should use results from models used in previous tasks to justify your answer).

9 Principal Component Analysis (PCA)

Use the Principal Component Analysis (PCA) method to perform feature selection in your dataset. Using the result of the feature selection performed with PCA, evaluate whether the models used previously can achieve better performance.

Compare the results obtained with those obtained in previous tasks (especially with the results from Task 5 - Feature Selection). What can you conclude about feature selection using PCA?

10 Reinforcement Learning (RL)

In this task, the goal is to design a system where Q-learning is used to sequentially select features for a classification model (must choose the two best previous models). The idea is to treat the feature selection process as a reinforcement learning problem, where an agent learns to choose which features to use to build the best classification model.

Problem Formulation:

- State: The state is a binary vector that represents the features selected so far. For instance, if the dataset has 14 features, the state is a 16-dimensional vector, where 1 means the feature is selected, and 0 means it is not.
- Actions: The action space is the set of all remaining features that haven't been selected yet. The agent can either select a feature or decide to stop selecting features (when it has enough information).
- Rewards:

  - A positive reward is given when the selection of features leads to an improvement in classification accuracy (after training a model with the current set of selected features).

  - A negative reward is given if the selected feature doesn't improve performance or leads to overfitting.

  - A penalty is applied for selecting too many features (to encourage simplicity and avoid overfitting).
- Goal: The agent's goal is to learn to select an optimal subset of features that maximizes the accuracy of the classifier while minimizing the number of features used.

11 Submissions

A notebook with answers to the proposed tasks. The notebook is .ipynb by default. Any other format must be easily readable. Please take care with the following:

- The steps taken must be succinctly described (through comments in the code or text cells in the notebook)
- The results must be summarized as much as possible.

11.1 Groups

- Assignments are submitted by groups of 3 or 2 students. Different elements may have different grades based on the contribution distribution and interactions about the assignment.
- Code of Conduct
  - All the materials used and consulted must be credited in the work as references.
  - All students should know the Disciplinary Regulations for Students of Polytechnic Institute of Porto (https://dre.pt/dre/detalhe/despacho/4103-2013-2301392)
- It is mandatory the Github version control tool. Each group must share the repository with PL teacher.

11.2 Deadline

There **two mandatory deliveries** of the work in Moodle:

- **2nd November**, intermediate delivery (25% final grade)
- **28th December**, final delivery (45% final grade)

Only submissions on the Moodle, before the deadline, will be considered to evaluation. Submissions after that date will not considered.

The name of the zip file should be: `APRAU_AAA_CCC_Num1_Num2_Num3.zip`, where: AAA is the teacher´s acronym, CCC the class and Numx the number of each student.

The presentation and discussion, mandatory for all group members, will be on a date to be scheduled by the PL teacher (cf. FUC APRAU course).