STUDY OF NON-PROBABILISTIC CLASSIFIERS

Author: Fontecha del Ser, Álvaro
Course: 103000897 - Machine Learning

1 Problem Description

The primary objective of this project is to implement various classification models to accurately categorize Spotify tracks based on their popularity. We aim to evaluate the performance of each non-probabilistic algorithm and examine how different feature subset selection strategies can influence their performance. To this end we have selected the following algorithms:

- **Gain Ratio Attribute Evaluation:** For our Univariate Filter FSS we have aimed to optimize the relevancy of the variables.
- **Correlation-based Feature Selection (CFS):** On the other hand, for Multivariate Filter FSS we hope to refine our selection by taking into account the minimization of redundancy.
- **Wrapper approach:** Our last most exhaustive search will be a wrapper method tested on each algorithm.

2 Introduction

The dataset we will be using is the Spotify Tracks dataset Pandya (2023), which has been extracted from the Spotify API. This dataset encompasses a carefully selected collection of 114,000 tracks, evenly distributed across a spectrum of 125 distinct genres.

The dataset's attributes include information about the artists, album names, track names, track genres, track duration, musical key, musical mode, tempo, time signature, and whether a track is marked as explicit or not.

We have followed notation and procedures as described in Bielza and Larrañaga (2021) and as imparted in 2023's fall course in Machine Learning.

Notation:

- **`Popularity`** is quantified using an algorithm that considers both the total number of plays a track has received and the recency of those plays.
- **`Danceability`** assesses how well-suited a track is for dancing, taking into account factors such as tempo, rhythm stability, beat strength, and overall regularity.
- **`Energy`** is a perceptual measure of the track's intensity and activity.
- The **`key`** of the track is represented using integers, mapping to pitches according to the standard Pitch Class notation. If the key is undetectable, the value is -1.
- **`Loudness`** corresponds to the overall volume of a track, measured in decibels (dB).
- **`Mode`** denotes the modality of a track. Major mode is indicated by 1, while minor mode is represented as 0.
- **`Speechiness`** identifies the presence of spoken words within a track.
- **`Acousticness`** provides a confidence measure regarding whether a track has acoustic characteristics.
- **`Instrumentalness`** predicts whether a track contains solely instrumental parts, excluding vocal content.
- **`Liveness`** indicates the likelihood of the presence of an audience in the recording.
- **`Valence`** measures the musical positiveness conveyed by a track, reflecting emotions such as happiness, cheerfulness, and euphoria.

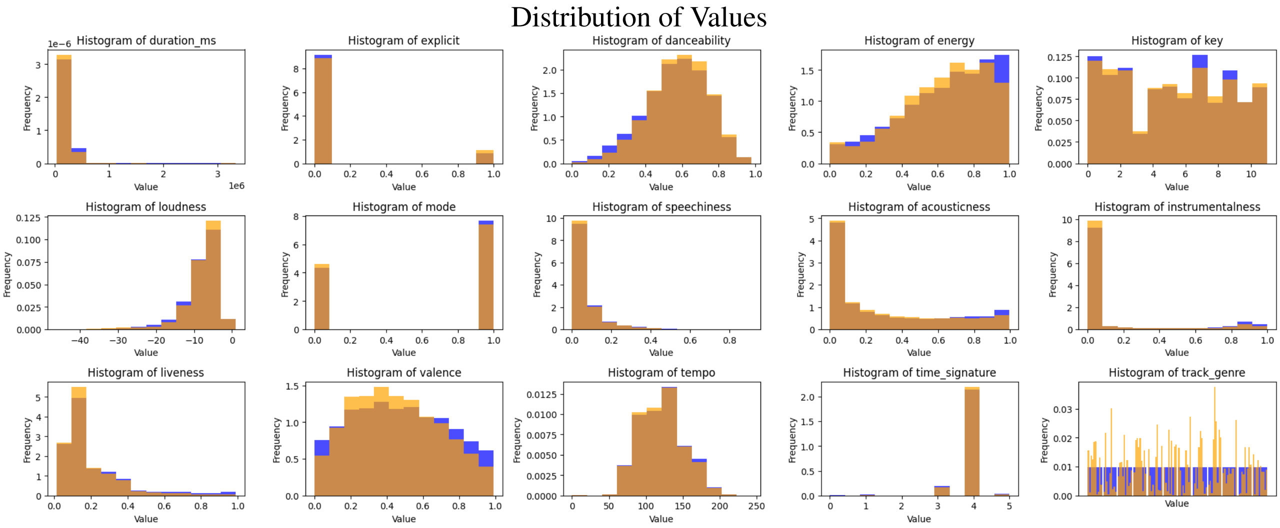

Figure 1: Distribution of variables across all data points (blue) and within the positive class instances (orange)

2.1 Exploratory Data Analysis

We have performed an initial study of the distribution of variables in our dataset. Preliminary analysis of the marginal distributions (Figure 1) reveals that the track_genre reveals important information about the class as it varies greatly within its nominal values. This preliminary analysis will aid us in comprehending posterior design decisions and model results.

Further studying the correlation relationships (Figure 2) we observe several correlated variables such as loudness and energy. Indeed we can infer a high number of decibels indicates an energic track. Equaly they are negatively correlated with an acoustic track. These variables will be prime candidates to be excluded in FFS if we take redundancy into account.

3 METHODOLOGY

For this project Python has been employed extensively for data preprocessing and exploratory analysis. For training the models and performing FSS as well as their evaluation, the software Weka Frank et al. (2016) was used.

3.1 Data Preprocessing

The initial dataset is remarkably clean, requiring minimal preprocessing with only a few null values that need to be handled.

An initial selection of variables is carried out, and certain attributes such as `track_id`, `artists`, `album_name`, and `track_name` are intentionally discarded. These attributes are excluded because they possess a large number of values and are presented in text format, which may not be suitable for the specific machine learning algorithms considered in this study.

Moreover, the attributes `key` and `track_genre` are subject to one-hot encoding, as they fall under the category of categorical variables with no inherent order or relationship.

Additionally, due to the inherent differences in magnitude among the various attributes, a normalization process is applied. This is especially important because certain machine learning models being considered in this study are highly sensitive to the differences in attribute scales.

3.2 Performance Evaluation

We will conduct an analysis of the performance of a binary classification problem. In this context, **accuracy** stands as the most commonly utilized metric for evaluating the effectiveness of an algorithm. Nevertheless, it

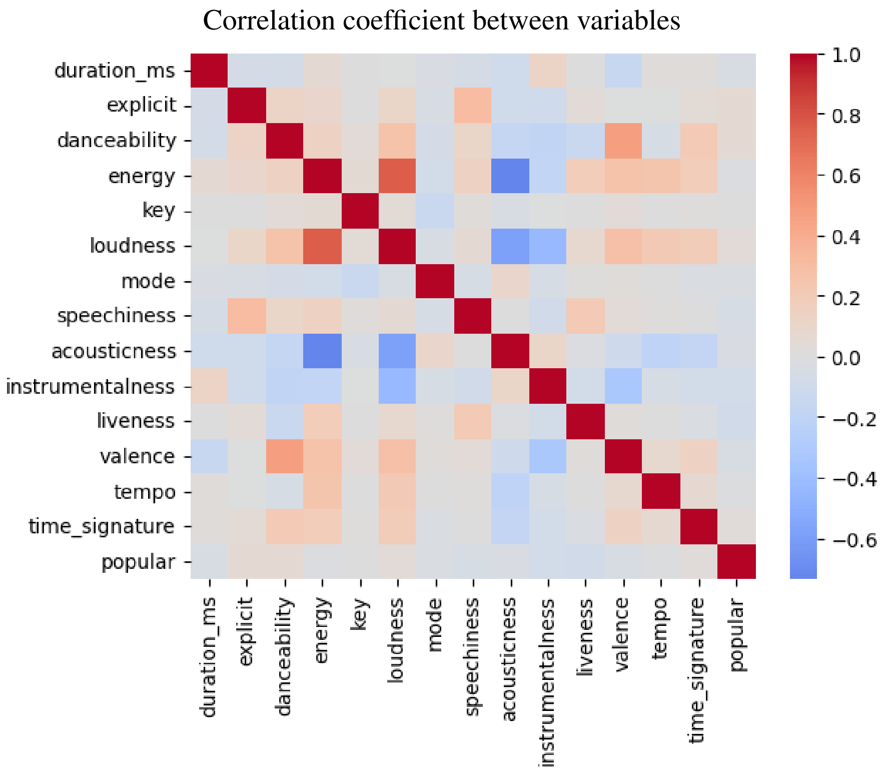

Figure 2: Correlation Heatmap

is worth noting that, within the framework of this particular problem, **recall** quantifies the algorithm's ability to detect popular songs. From an objective standpoint, such capability holds significance, as it provides a robust indicator of a new track's potential impact.

To incorporate the values of these two metrics, a widely acknowledged measure in the literature is employed – the **F$_1$ measure**. Being the harmonic mean of accuracy and recall, it is our preferred choice for comparing the performance of various models.

3.3 Honest Estimation Method

Given the substantial size of the dataset, there is no need for extensive resampling techniques. Instead, a 10-fold cross-validation approach is adopted to estimate the model's performance statistics. This methodology is favored for its capacity to provide unbiased estimates of the model's performance.

3.4 Classification Algorithms

Five distinct classification algorithms have been selected for this study. These include:

- **K-Nearest Neighbors (k-NN):** The k-NN algorithm serves as the initial analytical tool in our study. It is characterized by its intuitive nature, requiring minimal assumptions about the data, and providing an initial classification of data points based on a full feature set or a selected subset. However, the performance of k-NN is influenced by the choice of the parameter 'k,' which represents the number of nearest neighbors considered.

  In our approach, we adopt the IBk model, which allows for the optimization of the k parameter through cross-validation. We run an initial analysis with hold-one-out cross-Validation employed to select the optimal k value between 1 and 10. This parameter will be the one employed in our subsequent analysis.

  We acknowledge the sensitivity of the k-NN algorithm to irrelevant variables and anticipate improved performance following feature subset selection.

Since our primary objective is to assess the impact of feature subset selection (FSS), we do not consider the use of techniques involving either neighbor weighting or variable weighting. In practice, these methods can be seen as a form of feature selection themselves and would introduce confounding factors into our evaluation of FSS.

- **Rule Induction (RIPPER):** RIPPER, an iteration of IREP (Iterative Reduced Error Pruning) which adds a prunning phase, is a rule-based classification algorithm chosen for its interpretability and effectiveness in dealing with noisy data. RIPPER generates a set of if-then-else rules that represent decision boundaries within the dataset. These rules are makes the model easy to understand and interpret, particularly important for gaining insights into which decisions may contribute to track popularity in our case. Furthermore it may be the most useful algorithm in our case study for this reason.

- **Artificial Neural Network (MLP):** The Multilayer Perceptron (MLP) is a type of artificial neural network chosen for its ability to capture complex, non-linear relationships within the data. Given the diverse range of attributes describing Spotify tracks, an MLP can adapt to discover intricate patterns in the dataset. Its multiple layers of neurons allow it to model intricate interactions between features, making it a strong candidate for classification tasks. It Hotencodes our categorical attribute track genre, therefore obtaining a fine description of our dataset in a vector of 140 input neurons for the MLP.

  Due to computational constraints we have limited our training to $N = 200$ and 3 hidden layers which may reflect poorly on this method's accuracy.

- **Support Vector Machine (Polynomial Kernel):** Support Vector Machines (SVM) are widely appreciated for their effectiveness in handling high-dimensional data and finding non-linear decision boundaries. We chose to use SVM with a polynomial kernel to account for non-linearity in the dataset. Music popularity is a complex concept influenced by various acoustic and contextual factors, and a polynomial kernel can capture these intricate relationships. SVM, when used with the right kernel, can adapt well to complex data distributions, making it a robust choice for our classification problem.

- **Classification Tree:** C4.5 is a decision tree algorithm selected for its simplicity and interpretability. Decision trees are intuitive models that make classification decisions based on a series of attribute tests. C4.5 uses a top-down recursive approach to partition the dataset into subsets, and it selects the best attribute for each split. By implementing C4.5, we can generate a tree-like structure that visually represents the decision-making process. Similarly to RIPPER it proves to be particulary useful in our case study.

4 RESULTS

4.1 Feature Selection Subsets

We begin our discussion of the feature subsets selected with each method.

- **Univariate case:** In our Gain Ratio Attribute Evaluation, we have employed a ranker method to assess attribute relevance. This method ranks the attributes based on their informativeness by ranking the mutual information gain $\mathbb{I}(X_i, C)$ of the variables.

$$\mathbb{I}(X_i, C) = \sum_{x_i \in X_i} \sum_{c \in C} P(x_i, c) \cdot \log_2 (P(x_i, c))$$

  As this formula favors categorical features with many values (such as track genre) we opt for using the Gain Ratio instead: $\frac{\mathbb{I}(X_i,C)}{\mathbb{H}(X_i)}$ In this context, we have chosen to retain the top 10 attributes from a total of 15, establishing a metric threshold at a value of 0.002. This selection process allows us to focus on the most informative attributes for our analysis while reducing considerably the dimensionality.

- **Multivariate case:** For the Correlation-based Feature Selection, our initial set of variables has been significantly reduced, retaining only the ”track_genre” attribute. While this outcome may seem suboptimal, it is driven by the observation that the genre variable exhibits a high degree of correlation with the other

Table 1: Ranked Features according to Gain Ratio

Attribute	$\frac{\mathbb{I}(X_i,C)}{\mathbb{H}(X_i)}$
track_genre	0.025395
explicit	0.005037
liveness	0.004406
instrumentalness	0.004331
duration_ms	0.003015
valence	0.002992
danceability	0.002871
energy	0.00255
acousticness	0.00241
speechiness	0.002396
loudness	0.001822
time_signature	0.0018
tempo	0.000897
mode	0.000587
key	0.000355

features within each of its values (musical genres). It is plausible that the genre plays a pivotal role in predicting track popularity, acting as a summarizing variable that strongly influences the outcome. Nevertheless, it's worth noting that this particular subset maximizes the CFS metric:

$$f(S) = \frac{\sum_{i=1}^{k} \mathrm{cor}(\mathrm{X}_i, C)}{\sqrt{k + (k-1) \cdot \left(\sum_{i=1}^{k} \sum_{j=i+1}^{k} \mathrm{cor}(\mathrm{X}_i, \mathrm{X}_j)\right)}}$$

In some cases, a single feature can capture a significant portion of the predictive power of a model, making it the most influential factor. Given that CFS aims to minimize feature redundancy and maximize the relevancy of the selected features, it might conclude that track_genre alone is highly informative in predicting track popularity. This reduction in dimensionality can lead to more efficient and interpretable models while maintaining or even improving predictive performance.

- **Wrapper approach:** In the context of our study, the wrapper approach holds a special significance. We have made a deliberate choice to optimize our model selection process using the F_1 measure. To optimize our model selection, we employ a Best First search strategy, which aims to find the best subset of features iteratively. This strategy explores different combinations of features, evaluating their performance using said measure.

  We have chosen to utilize a 5-fold cross-validation process within the Best First search, which involves partitioning our dataset into five equally sized subsets. This cross-validation method helps ensure that our model's performance is reliable and unbiased. The combination of the Best First search and 5-fold cross-validation allows us to systematically evaluate and refine our feature subset, ultimately leading to a model that exhaustively searches for the best feature subset.

Table 2: Attribute Names

	track_genre	explicit	liveness	instrumentalness	duration_ms	valence	danceability	energy	acousticness	speechiness	loudness	time_signature	tempo	mode	key
Gain Ratio	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
CFS	✓														
Wrapper (K-NN)	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓		
Wrapper (RIPPER)	✓	✓	✓	✓	✓			✓	✓	✓					
Wrapper (MLP)	✓	✓	✓	✓	✓			✓		✓					
Wrapper (SVM)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
Wrapper (C4.5)	✓		✓	✓		✓	✓	✓	✓	✓			✓		

Table 3: K-NN Algorithm Performance Measures

	Original Variables	Univariate FSS	Multivariate FSS	Wrapper FSS
Accuracy	0.777	0.777	0.759	0.754
Recall	0.787	0.787	0.783	0.755
F_1 Measure	0.781	0.781	0.753	0.754

5 DISCUSSION

The outcomes unequivocally favor Feature Selection (FSS). Within each model, the univariate filtering method demonstrates an at least analogous performance to using the original variables. The results from Correlation-Based Feature Selection (CFS) are noteworthy. They suggest a selection of only the genre, suggesting a substantial correlation between track genres and other descriptive variables, possibly attributed to Spotify's algorithm utilizing these variables for encoding track genres. This approach simplifies the dataset, enhancing the accuracy in simpler models like RIPPER. However, in more intricate models, some information loss occurs, leading to reduced accuracy despite significantly lower computational costs. Notable consequences include Multi-Layer Perceptron (MLP), which only requires one hidden layer to process this singular variable, effectively acting as a linear model. In the case of RIPPER, the set of rules comprises a disjunction of genres. C4.8 follows a similar pattern, having only 114 leaves and 1 parent, but this does not result in improved metrics as it essentially represents a genre selection. If the goal is to exclusively optimize the F_1 measure, such a drastic reduction in

Table 4: RIPPER Algorithm Performance Measures

	Original Variables	Univariate FSS	Multivariate FSS	Wrapper FSS
Rules	7	5	14	6
Accuracy	0.756	0.752	0.759	0.764
Recall	0.763	0.763	0.783	0.785
F_1 Measure	0.672	0.673	0.751	0.775

Table 5: MLP Measures

	Original Variables	Univariate FSS	Multivariate FSS	Wrapper FSS
Accuracy	0.764	0.764	0.761	0.781
Recall	0.787	0.787	0.784	0.787
F_1 Measure	0.754	0.760	0.760	0.783

Table 6: SVM Performance Measures

	Original Variables	Univariate FSS	Multivariate FSS	Wrapper FSS
Accuracy	0.763	0.763	0.763	0.763
Recall	0.786	0.786	0.786	0.786
F_1 Measure	0.753	0.753	0.753	0.753

features proves ineffective.

Wrapper approach is as expected more accurate than the rest. Howe Among the models, IBk initially outperforms the rest. With a more advanced FFS, MLP and SVM yield higher scores, but this is contingent on increasing the complexity and computational costs of the algorithms. A concise interpretation may be that these variables pertain to subjective concepts, particularly genre, and are therefore inherently fuzzy concepts.

Regarding popularity characterization, it is evident that certain genres are more favored than others. Similarly, within a specific genre, a set of characteristics dictates how to achieve popularity. This is effectively modeled by the K-Nearest Neighbors (K-NN) algorithm, as similar tracks within the same (popular or not) genre tend to cluster together, with the most popular tracks resembling each other. Future work could explore weighting this variable to encourage that conceptual behaviour. In its particular case FFS does not increase accuracy, and it seems the model performs better with higher number of variables.

RIPPER proves especially valuable in this problem case by extracting a set of rules or guidelines for creating new tracks that optimize popularity. Nevertheless, the effectiveness of these rules may vary between genres and warrants further investigation. It is worth noting that RIPPER, while the least accurate, is the algorithm most influenced by FSS.

Similarly, C4.8 operates in a similar manner, but it does not generate a set of explicit rules, making it more challenging to draw conclusions. Nonetheless, it is a relatively simple algorithm that performs admirably while maintaining a low computational cost.

Both MLP and SVM deliver the anticipated solid performance and offer accurate predictions for popularity. Within these models, it is evident that FSS does not enhance accuracy fixing it, but it significantly reduces their computational costs. In the case of SVM its Wrapper FSS selected the same criterion as in the Univariate case. However it seems even between Univariate and Multivariate the accuracy is not improved at all

Table 7: C4.8 Performance Measures

	Original Variables	Univariate FSS	Multivariate FSS	Wrapper FSS
Tree size	5509	2043	115	1831
Accuracy	0.779	0.782	0.759	0.784
Recall	0.793	0.798	0.783	0.799
F_1 Measure	0.782	0.785	0.753	0.787

6 CONCLUSION

References

Bielza, C. and Larrañaga, P. (2021). *Data-Driven Computational Neuroscience: Machine Learning and Statistical Models*. Cambridge University Press.

Frank, E., Hall, M., Trigg, L., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2016). Weka: Data mining software in Java. `https://www.cs.waikato.ac.nz/ml/weka/`. Version 3.9.4.

Pandya, M. (2023). Spotify Tracks Dataset. `https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset`.