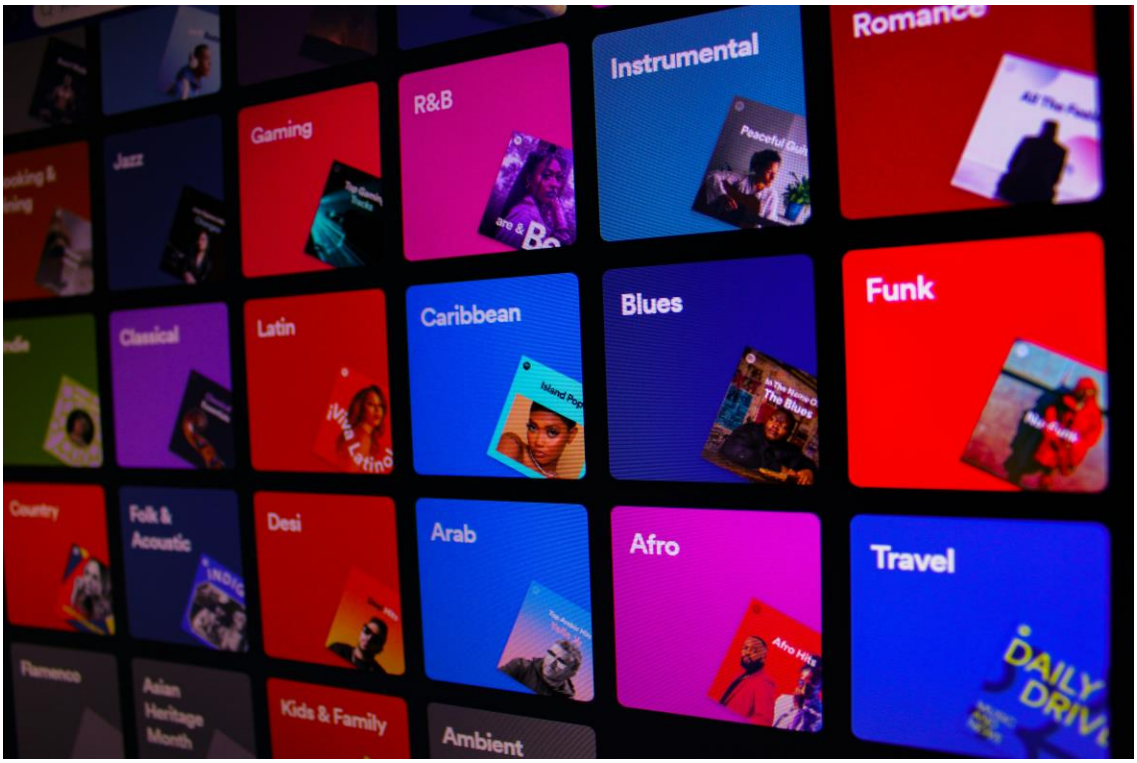
# Spotify Genre Classification using Machine Learning

**By Maya Geva**

**Machine Learning Module Project**

**2025**



## 1. Introduction

This project explores how machine learning can be applied to classify music genres based on Spotify's audio features dataset. The goal is to identify which audio characteristics best differentiate between genres and to develop predictive models capable of classifying songs accordingly.

I chose this topic after a debate at work about whether a certain song was rock or pop - a surprisingly complex question. When I saw Spotify among the available datasets, it immediately fit. I also realized that genre classification models could benefit AI music products, helping to pre-label melodies by genre.
The potential value: saving human labelling time, improving data quality, and reducing classification errors.

## 2. Objectives

The main objectives of this project are to explore how machine learning can classify songs by genre using Spotify's audio features and to identify which characteristics best distinguish between musical styles. Specifically, the project aims to:

- Analyse and visualize relationships between audio features and genres.
- Build and evaluate machine learning models for automatic genre prediction.
- Optimize model performance through feature engineering and fine-tuning.
- Demonstrate the potential value of genre-classification models in AI music applications by improving tagging efficiency and data quality.

## 3. Project Design & Methodology

3.1 The data journey begins with **data preparation**. The dataset, obtained from Kaggle, contained 32,833 rows and 23 columns and was relatively well-organized in a single table, so there was no need to merge multiple datasets.
During this stage, I cleaned text fields, normalized casing and punctuation, and extracted temporal attributes such as release_year and release_month. I also reduced rare categories in text-based columns by grouping them under an "other" category to simplify the analysis.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32833 entries, 0 to 32832
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   track_id                  32833 non-null  object
 1   track_name                32828 non-null  object
 2   track_artist              32828 non-null  object
 3   track_popularity          32833 non-null  int64
 4   track_album_id            32833 non-null  object
 5   track_album_name          32828 non-null  object
 6   track_album_release_date  32833 non-null  object
 7   playlist_name             32833 non-null  object
 8   playlist_id               32833 non-null  object
 9   playlist_genre            32833 non-null  object
 10  playlist_subgenre         32833 non-null  object
 11  danceability              32833 non-null  float64
 12  energy                    32833 non-null  float64
 13  key                       32833 non-null  int64
 14  loudness                  32833 non-null  float64
 15  mode                      32833 non-null  int64
 16  speechiness               32833 non-null  float64
 17  acousticness              32833 non-null  float64
 18  instrumentalness          32833 non-null  float64
 19  liveness                  32833 non-null  float64
 20  valence                   32833 non-null  float64
 21  tempo                     32833 non-null  float64
 22  duration_ms               32833 non-null  int64
dtypes: float64(9), int64(4), object(10)
memory usage: 5.8+ MB
```

3.2 In the **Exploratory Data Analysis (EDA)** stage, I generated visualizations such as violin plots, boxplots, and correlation heatmaps to explore relationships between features. I began by analyzing the distributions and relationships of genres and subgenres to understand

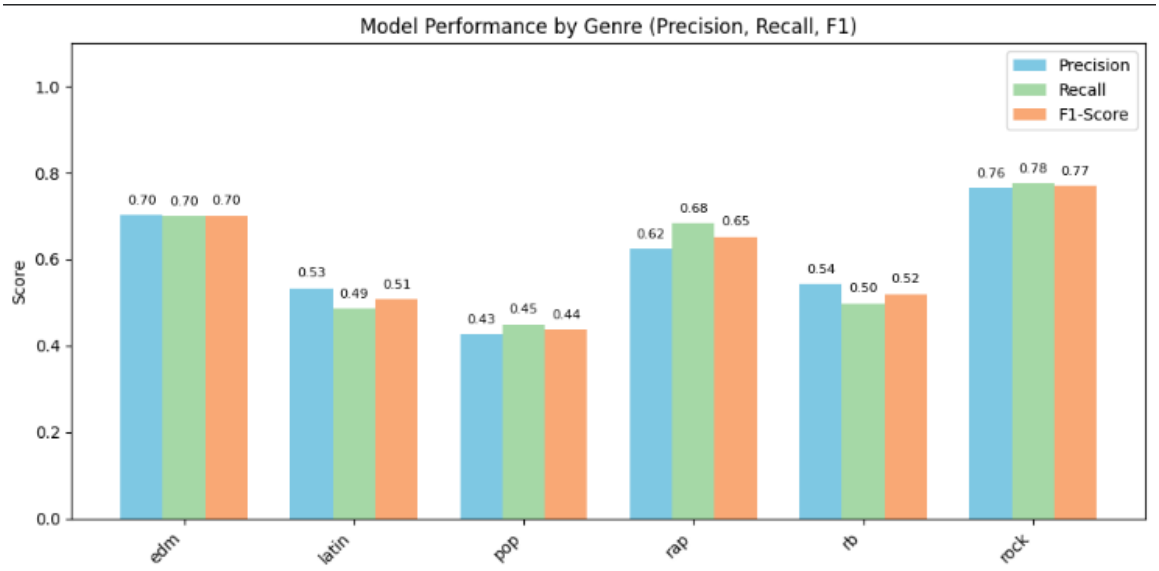
how they were connected before deciding which of these features to use as target in the modeling phase.
Significant correlations included positive relationships between **energy** and **loudness**, and negative correlations between **acousticness** and **energy**.
Non-parametric statistical tests such as **Spearman** and **Kruskal–Wallis** confirmed that most audio features differ significantly across genres.

3.3 In the **Data Cleansing and Outlier Treatment** stage, I detected outliers using the **IQR method** and evaluated their impact on feature distributions and correlations with playlist_genre. **Winsorization** was applied only where outliers distorted distributions without affecting correlations- specifically for acousticness, liveness, duration_ms, loudness, and tempo.

3.4 In the **Feature Engineering and Selection** stage, I created new numeric and interaction features, including ratio-based metrics, temporal features, and composite measures. Numeric variables were standardized using **StandardScaler**.
Feature selection combined statistical and model-based methods - **ANOVA F-test** and **L1 Logistic Regression** -along with **Model Committee Voting** across Logistic, SVM, Gradient Boost, and Random Forest models. Comparing **Union** and **Intersection** feature sets, the **Union** approach achieved the best performance with an F1-score of approximately **0.59**.

## 4. Models

In the **Model Selection and Fine-Tuning** stage, the dataset was split into training, validation, and test sets (≈64% / 16% / 20%). Several algorithms were optimized using **GridSearchCV**, including Logistic Regression, SVM, Random Forest, Gradient Boost, AdaBoost, and XGBoost.
The **XGBoost_CV** model achieved the best results with a **Macro-F1 of 0.60** on the test set. It performed best for **Rock**, **EDM**, and **Rap**, while results for **Pop**, **Latin**, and **R&B** were comparatively weaker.



## 5. Deployment

The deployment phase envisions integrating the trained genre classification model into AI-driven music generation or analysis systems such as Suno or AIVA.
The model could serve as a pre-tagging engine that automatically assigns genres to newly generated or uploaded tracks, improving labeling efficiency and data consistency.
It could be deployed as an API service that receives audio features and returns a predicted genre label, supporting both batch and real-time processing.
Cloud-based deployment (e.g., AWS or GCP) would enable scalable performance, version control, and continuous retraining as new music data becomes available.

## 6. Conclusion

This project demonstrates the use of machine learning for music genre classification using Spotify's dataset. Through careful data preprocessing, visualization, and model tuning, it highlights the relationships between musical attributes and genre distinctions.

**Future Potential**

Over time, the model could be extended toward more advanced capabilities:

- **Multi-genre detection:** allowing a song to belong to multiple genres (e.g., 40% Rock, 60% Blues).
- **Style evolution tracking:** identifying stylistic changes across albums or over time.
- **Genre recommendation engine:** automatically suggesting similar songs based on genre profiles.