
SIMON FRASER
UNIVERSITY

# Numerical Feature-Based Music Genre Classification Using Supervised Learning Techniques: Utilizing K-Nearest Neighbours

Submitted to:
**Morteza Badali**

Simon Fraser University
Faculty of Applied Sciences
CMPT310:
Introduction to Artificial Intelligence

Submitted by:
**Jeongmin Jang**
**Aung Htet Khine**
**Clark Darneill Noveros**
**Thant Hayman Thway**

# I. Background

In recent years, listening or *streaming* has changed drastically and has adapted into the technologically-dense world and socio-economic standards. From what started as a mere hominid ability and a derivation of speech; to emit sounds through vibrations from the vocal chords. Anthropogenic motor impulses that translated into rhythm (Montagu, 2017) have evolved into a complex, technologically driven and striving industry. Music has evolved drastically and is constantly shaped by the trend of the *AI movement*, with major companies striving to integrate artificial intelligence into their products and services, as well as altering their operations with the use of AI for better delivery and faster turnover. Consequently in 2023, Spotify released DJ X, voiced by Spotify's Head of Cultural Partnerships, Xavier "X" Jernigan. Powered by OpenAI, this new virtual DJ knows their user personally, curating a mix of different songs, from old favourites, newly discovered, and on rewind songs. This AI DJ also has the ability to take recommendations as prompts and will deliver moving forward. A study by Mukherjee et. al (2025) that during 2022, Spotify contributed as the top subscribed music streaming provider with 30.5% domination across the global market against eight (8) more other competitors. However, Mukherjee and colleagues (2022) analyzed and collected subreddit comments and remarks, with a dataset of 92 posts and 1,442 comments. Moreover, their findings consisted of 34% negative remarks, 32% percent of those were from poor song selections from DJ X. Top word queries from that section were *songs, dj, playlist, discovery, repeat*, which intuitively can suggest that the AI DJ were not up to the users standards, coming off with poor song suggestions that may be too repetitive, or users want to discover new music. The groups project offers a simple yet effective supervised genre classification model utilizing KNN. Unlike huge streaming platform services, the group's approach focuses on transparent genre classification from multiple audio features and systematically evaluates performance using 10-fold cross-validation. Furthermore, with the help of the aforementioned approaches, the model having a high accuracy indicates that the model generalizes well and is reliable in classifying data across unseen tracks. It is known that generalizability is a huge factor in creating models, since it is the modulator and median for both overfitting and underfitting of data.

# II. System Explanation

## II.a Describing the System

The project is a music genre classification system that predicts the genre of a song, based on extracted audio features. The model uses a supervised learning technique, K-Nearest Neighbor (KNN) trained on GTZAN music dataset which is a collection of 30 second audio tracks from 10 different genres. With this, 10, 30, and 60 feature subsets are used to explore how the number of features affects the performance of the model. The 60 feature set was constructed by using mean and variance for each original feature, while for the 30 feature set it used median values. The 10 features set contains further reduced subsets selected from the 30 feature set. These subsets were created in R and exported as CSV files for analysis in Python.

## II.b AI Methods

A recent work by Mycka and Mandziuk have stated that genre classification is often tackled by a special type of neural network called Conditional Neural Networks or Masked Condition Neural Networks; with this, some papers were also said to use Convolutional Recurrent Neural Networks. On the other hand, with datasets that are not too huge, and using handcrafted feature extraction and dependencies, commonly used Machine Learning models have been used, in this case the K-Nearest Neighbours.

The KNN is a lazy learner and is versatile, which makes it a good choice for the group, who has recently just started out with machine learning. It allowed the group to run some analysis to determine the best K value to use for each of the subsets. The group chose this learning model since it is known to clump-up data that has common features, which can be applied to clumping-up music with the same genre, and visually it will show labels that are grouped together. However, it still boils down to accuracy and features used. Therefore, since the subset is not that large, the model is confined to the subset, it is known that generalizability must be accommodated for new or unseen values.

K-Fold Cross Validation is applied by the group in lieu with the KNN, in this case a 10-fold cross validation. The cross validation estimates the accuracy and how well the model performs with unseen data, since the group used an 80-20 train-test split, 10-fold cross validation is applied.

## II.c Pipeline

1.Data Input

The initial stage of this machine learning pipeline involves data acquisition. Cleaned feature datasets , provided as 10, 30 and 60 feature subsets served as input. Each subset comprises a fixed-length feature vector for individual songs, alongside their corresponding genre labels, which act as the target variable for prediction .

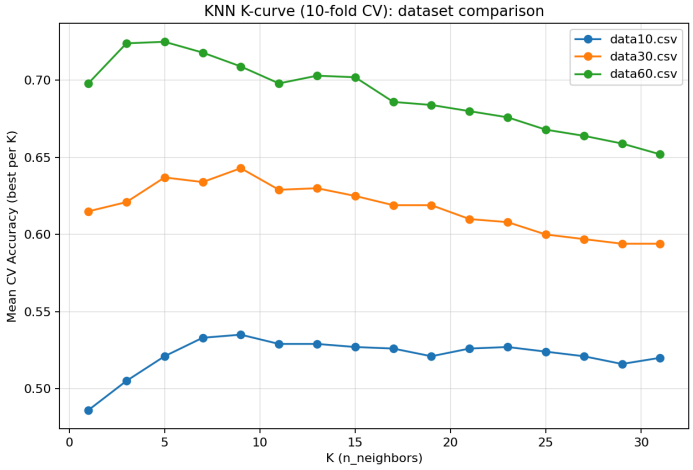
2. Preprocessing

Prior to modeling, the data undergoes essential preprocessing steps to ensure its suitability and quality. This includes the removal of non-feature columns and the conversion of all feature values to a numeric format. Missing values, if present, are addressed by imputing them with the median of their respective column. Finally, a standardscaler is applied to the feature matrix to normalize all features to a consistent scale, characterized by zero mean and unit variance which is crucial for effective model training.

3. Model Training & Evaluation

Following preprocessing, the pipeline moves into model training and hyperparameter optimization using a K-Nearest Neighbors (KNN) classifier. A rigorous approach is employed, involving a comprehensive hyperparameter search combined with cross-validation. Specifically, 10-fold cross-validation is performed on each data set to tune critical parameter such as K value,

the distance metric (e.g. Euclidean vs Manhattan) and the weighting scheme (uniform vs distance-based). A grid Search is conducted over odd K values ranging from 1 to 31 to prevent tie votes in classification. For each candidate setting, the model is evaluated across the folds and the combination yielding the highest cross-validation accuracy is selected. To ensure an unbiased evaluation of the final model's performance on the unseen data, 20% of the dataset is set aside as an independent hold-out test set. All model decisions, including the selection of optimal K and the distance metric are exclusively based on the training and cross-validation data.



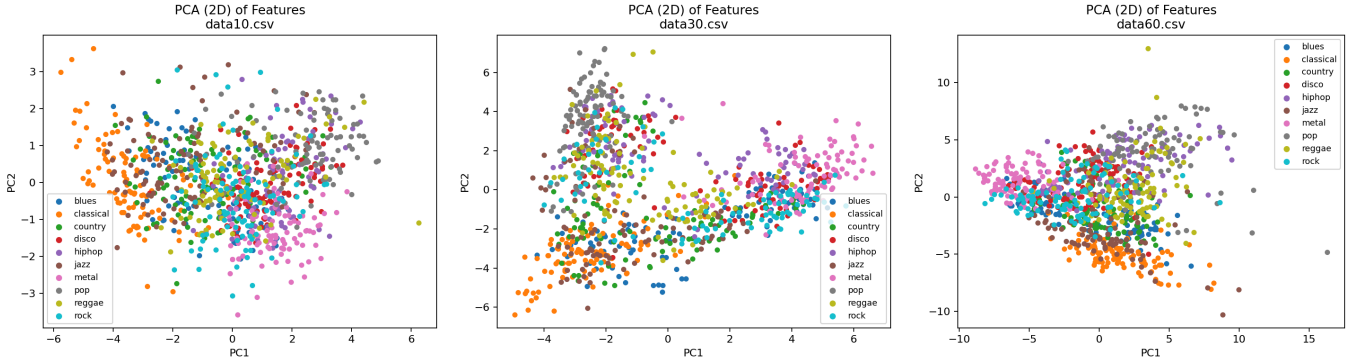

The subsequent phase focuses on prediction and evaluation. Utilizing the best-found KNN model with its optimized parameters, genre predictions are generated and its performance is rigorously assessed. Key metrics measured include overall accuracy, representing the fraction of correctly classified songs and the macro F1-score. The macro F1-score is particularly important as it provides the average F1-score across all genres, treating each genre equally, thereby ensuring that model performance is balanced and not disproportionately influenced by larger classes.

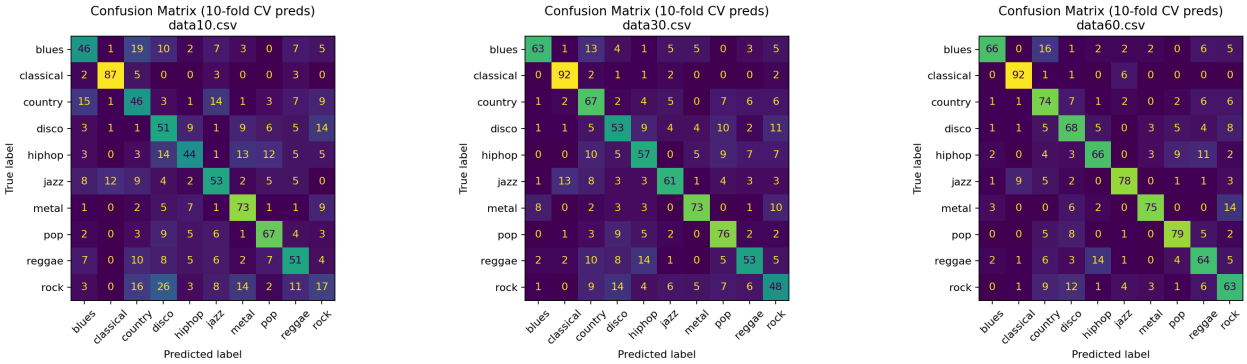
4. Visualization and Output Generation

This is performed to facilitate interpretation and understanding of the model's result. Confusion metrics are plotted as labeled heatmaps for each feature set, allowing for a visual comparison of prediction errors and identification of commonly confused genres. The resulting PCA scatter plots illustrate how songs cluster by genre in the feature space, potentially revealing clearer genre separation with higher-dimensional data.

Additionally, a K-curve is generated for each feature set, plotting cross-validation accuracy against different K values to show the impact of K on performance. A bar-chart summarizing accuracy versus the number of features is also produced, highlighting the effect of feature count on classification accuracy. The ultimate output of this pipeline consists of the

trained, optimized KNN model, capable of predicting genres for new song features and a comprehensive evaluation report detailing its performance across various feature subsets.




## II.d Limitations

There are a few limitations that hindered the development of the model but not to the extent that the model did not perform up to its expectations. KNN is a fairly simple model, every feature contributes equally after scaling, however since the data was cleaned of irrelevant features. Other models are known and capable of automatically learning which features matter the most, the group opted to experiment with feature set sizes and scaling the features. The next limitation was reducing the feature spaces, notably the 10 feature subset was derived from the 30 feature subset which led to substantial feature loss, shown in the lower accuracy for the 10 feature subset.

The group used a dimensionality reduction technique, Principal Component Analysis, to compress features into 2 dimensions. A smarter PCA-based reduction could potentially preserve more information.

Exploration in using other, more complex classifiers could improve accuracy and handle larger and more complex data sets versus what the groups KNN model accomplished. Lastly, integration into existing software of music streaming platforms, it would be better if the model is seen to perform in line with the consumer based AI DJ.

**2. Feature Table**

| **Description** | **Platform** | **Completeness** | **Code** | **Author** | **Notes** |
|---|---|---|---|---|---|
| Data Acquisition & Feature Processing | **Kaggle** | **5** | **R** | Aung H Khine, Clark Noveros | Used mean & variance for 60 features, median for 30 features and selected feature for 10 features |
| Data Cleaning & Normalization | **Local** | **5** | **R** | Aung H Khine, Clark Noveros | Cleaned dataset by removing irrelevant features |
| Feature Dimensionality | **Local** | **5** | **R** | Aung H Khine, Clark Noveros | Successfully created feature subsets with 10, 30, and 60 dimensions. |
| KNN Model Implementation | **Local** | **5** | **Python** | Aung H Khine, Clark Noveros, Jeongmin Jang, Thant H Thway | Implemented using sklearn. |
| 10-Fold Cross-Validation | **Local** | **5** | **Python** | Aung H Khine, Clark Noveros, Jeongmin Jang, Thant H Thway | Evaluated using sklearn. Achieved best CV accuracy of 72% for 60 features subset, 64% for 30 features subset and 53% for 10 features subset. |
| Visualization | **Local** | **5** | **Python** | Jeongmin Jang, Thant H Thway | Created PCA chart and best K graph using matplotlib |
| Evaluation & Matrix | **Local** | **5** | **Python** | Aung H Khine, Clark Noveros, | Used accuracy score and f1 score for evaluation. |

| | | | | | |
|---|---|---|---|---|---|
| | | | | Jeongmin Jang, Thant H Thway | Created a confusion matrix to visualize the evaluation. |

## 3.External Tools & Libraries

Programming Languages & Environment

Data preprocessing was done in R (for data cleaning and subset creation), and model training & analysis was done in Python. The Python environment included a Jupyter Notebook for interactive analysis and script development. The operating system was a standard 64-bit environment (e.g., Windows 10 or Linux) capable of running R and Python. No special hardware (GPU) was required, as KNN and dataset size were manageable on CPU.

Python Libraries

The project heavily relied on open-source Python libraries:

- **Scikit-learn** - machine learning framework used for the KNN classifier, grid search (GridSearchCV), cross-validation, and performance metrics. The KNN implementation from scikit-learn was used directly (no custom re-implementation of the algorithm), and we reused its built-in methods for scaling (StandardScaler), model evaluation (accuracy_score, etc.), and visualization (ConfusionMatrixDisplay for plotting confusion matrices).
- **Pandas** - used for data manipulation and analysis, such as reading CSV files into DataFrames, cleaning data, and storing results. For example, we used pandas to create summary DataFrames of results and to manage the grid search output.
- **NumPy** - used implicitly through pandas and scikit-learn for numerical computations and array structures. NumPy arrays underlie the feature matrices and make the distance calculations efficient.
- **Matplotlib** - used for creating all the plots (confusion matrices, PCA scatters, line plots for K vs accuracy, and bar charts). The plotting code customized titles, labels, legends, and annotations for clarity.

**R Libraries**

In the R preprocessing stage, common data handling packages like tidyverse (dplyr, tidyr) may have been used to aggregate and clean features. For instance, the team likely used R to compute means/variances per group and handle any data issues before export. (The exact R packages used were not listed, but base R or tidyverse tools were sufficient for these tasks.)

**Open-Source Code Reuse**

Apart from using standard library functions, the project did not copy any specific code from external sources. All modeling code (in Python) was written by the team, leveraging scikit-learn's API. The Kaggle dataset came with some feature extraction code (provided by dataset authors), but our team constructed the feature subsets independently. Any external code referenced (such as Kaggle notebooks or tutorials) was used only as guidance and not directly incorporated.

**Code Submission Instructions**

The full project source code and data are included in the submitted ZIP archive. To review or run the project, please refer to the following contents:

- **Source Code:**

  The Python script knn_compare2.py contains all the machine learning code (data loading, KNN model training, cross-validation, evaluation, and plotting). An R script was used for data preprocessing (e.g., aggregating features into 60/30/10 dimensions), it is also included in the archive as an .R file.

- **Datasets:**

  The cleaned feature subset files (data60.csv, data30.csv, data10.csv) are provided. These CSVs can be used with the Python script to replicate the experiments. (The original raw audio dataset from Kaggle is not included due to size.)

- **README**

  The archive includes a README file that explains how to set up a Python virtual environment, install the required packages (pandas, numpy, matplotlib, scikit-learn), and run knn_compare2.py with the 10, 30, and 60 feature datasets. It also summarizes what the script does and where the output files are saved.

- **Output Files**

  When you run knn_compare2.py, it creates a folder called knn_output/ and saves all result figures there. These include confusion matrix images (*_confusion.png), PCA scatter plots (*_pca.png), and K–accuracy curves (*_kcurve.png) for each feature subset. These are the same types of figures shown in our project poster and used in our analysis.

By installing the listed libraries and running knn_compare2.py with the provided CSV files, you should be able to reproduce our results.

References:

Montagu, J. (2017). *How music and instruments began: A brief overview of the origin and entire development of music, from its earliest stages. Frontiers in Sociology, 2.* https://www.frontiersin.org/journals/sociology/articles/10.3389/fsoc.2017.00008/full

Mukherjee, A., Chang, H., & Wibowo, J. (2025). *Three lessons from Spotify's personalization paradox.* SSRN. https://www.researchgate.net/publication/394112562_Three_Lessons_from_Spotify%27s_Personalization_Paradox

Stackpole, B. (2022, May 26). *How Wayfair and Spotify use machine learning to engage customers.* MIT Sloan School of Management. https://mitsloan.mit.edu/ideas-made-to-matter/how-wayfair-and-spotify-use-machine-learning-to-engage-customers

Weng, R (n.d.). Comparison Between Two Algorithms in Music Genre Classification. https://www.scitepress.org/Papers/2024/128297/128297.pdf?utm_source=chatgpt.com

Mycka, J & Mandziuk, J (2024). *Artificial intelligence in music: recent trends and challenges. Neural Computing and Applications.* https://link.springer.com/article/10.1007/s00521-024-10555-x