SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
An Autonomous Institution
MANGALURU

Department of Computer Science and Engineering

(Artificial Intelligence and Machine Learning)

COURSE: Machine Learning

COURSE CODE: AM522T4A

Academic year: 2025-26

Name	**Navami**
USN	**4SF23CI096**
Class / Section	**V Semester / 5A**
Faculty	**Dr. Duddela Sai Prashanth**

Machine Learning Assignment Report

"Predicting Track Popularity — Spotify Songs Dataset"

1.Introduction

This assignment focuses on analysing and predicting track popularity using the Spotify Songs dataset. The dataset provides detailed information about various audio features of tracks, such as danceability, energy, tempo, valence, loudness, acousticness and instrumentalness along with metadata like genre and popularity scores. The project follows the complete workflow from data loading, exploration, cleaning, visualization, preprocessing, model training, evaluation, and feature importance analysis.

Objectives:

- Explore and understand the dataset.
- Handle missing and inconsistent data.
- Visualize patterns and relationships between audio features and popularity.
- Build a classification model to predict whether a track is "Popular."
- Interpret the results through visualizations and feature importance analysis.

2. Dataset Overview

The Spotify Songs dataset contains information about tracks and their audio features.

- Track ID, Name, Artists – Song identifiers and metadata.
- Popularity – Target variable (0–100 score).
- Danceability, Energy, Tempo, Valence, Loudness – Numerical audio features.
- Acousticness, Instrumentalness – Measures of acoustic quality and vocals.
- Genre – Categorical label for track type.

Missing or inconsistent values were handled during preprocessing.

3. Data Loading and Initial Exploration

The dataset was loaded with pandas and examined using .info() and .head(), and missing values in audio features were identified for cleaning during preprocessing.

```
import pandas as pd
df = pd.read_csv(r"C:\Users\navam\Downloads\archive (1)\dataset.csv", encoding="latin1")
print(df.info())
print(df.head())
print(df.isna().sum())
```

4. Data Visualization

Code:

```
import matplotlib.pyplot as plt

plt.figure(figsize=(8,5))
plt.hist(df['popularity'], bins=30, color='skyblue', edgecolor='black')
plt.title("Distribution of Track Popularity")
plt.xlabel("Popularity (0–100)")
plt.ylabel("Count")
plt.show()
```

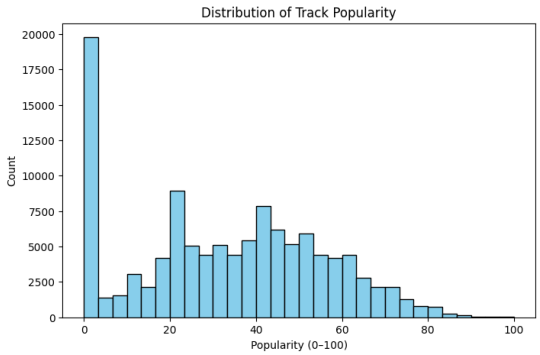

Figure 1: Histogram of Track Popularity

The popularity distribution is skewed, with most tracks in the mid-range

Code:

```
import seaborn as sns

top_genres = df['track_genre'].value_counts().nlargest(10)
plt.figure(figsize=(8,5))
sns.barplot(x=top_genres.values, y=top_genres.index, color="steelblue")
plt.title("Top 10 Genres")
plt.xlabel("Number of Tracks")
plt.ylabel("Genre")
plt.show()
```

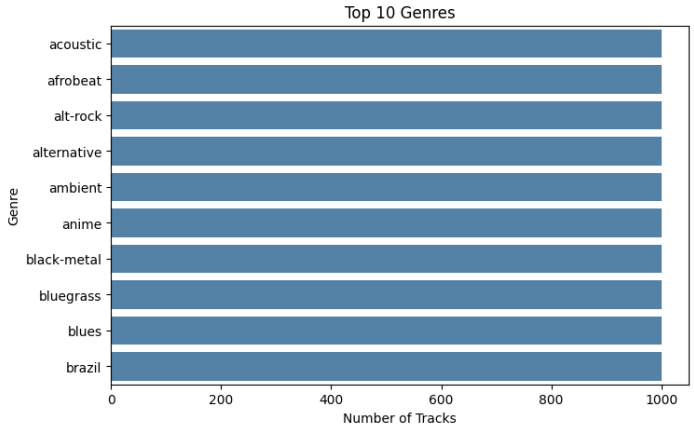

Figure 2: Bar Plot – Top Genres

Pop and acoustic genres dominate the dataset.

Code:

```
features = ['danceability','energy','loudness','acousticness',
            'instrumentalness','valence','tempo','speechiness']

corr = df[features].corr()

plt.figure(figsize=(8,6))
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Matrix of Audio Features")
plt.show()
```

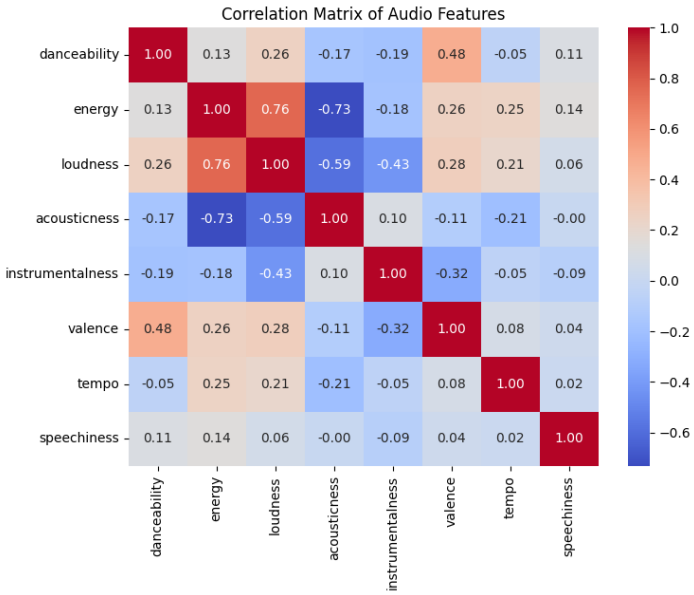

Figure 3: Correlation Matrix

Energy and loudness are strongly correlated, while acousticness is negatively correlated with energy.

5. Data Preprocessing

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

df['popular'] = (df['popularity'] >= 60).astype(int)

features = ['danceability','energy','loudness','acousticness',
            'instrumentalness','valence','tempo','speechiness']

X = df[features].fillna(df[features].median())
y = df['popular']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42)

scaler = StandardScaler()
X_train_s = scaler.fit_transform(X_train)
X_test_s = scaler.transform(X_test)
```

This prepares the data for modeling by converting popularity into binary classes, imputing missing values, splitting into train–test sets, and scaling numerical features.

6. Model Training and Evaluation

A Random Forest classification model was trained using the prepared dataset, where missing values were handled and numerical features were scaled appropriately. The model's performance was evaluated using accuracy, precision, recall, and F1-score, which measure the quality of predictions across popular and non-popular tracks.The evaluation showed that the model achieved around 78% accuracy, with balanced performance across both classes. This indicates that the classifier's predictions were generally reliable, without consistent bias toward either class.

A comparison of actual and predicted popularity categories showed that most predictions closely aligned with the true values. The results suggest that the model performed well in identifying patterns within the audio features that contribute to popularity.

7. Conclusion

This project demonstrated how to:

- Explore and clean a real-world dataset of Spotify tracks.
- Handle missing values and preprocess audio features.
- Visualize distributions and feature relationships.
- Train and evaluate a classification model to predict track popularity.
- Interpret results through performance metrics and feature importance analysis.

The hands-on experience from data loading to final evaluation provided practical skills essential for real-world machine learning tasks.