Taif University
College of Computers and Information Technology
Computer Engineering Major

جامعة الطائف
كلية الحاسبات وتقنية المعلومات
تخصص هندسة حاسب

MACHINE LEARNING PROJECT

Predicting Song Popularity on Spotify Using Machine Learning

Instructor: Dr. Nada Khamis Al-Tuwairqi
Course: Machine Learning
Section/Group: 4232
Submission Date: 16/5/2025

Reem Fawaz Abdullah Alqethami
441053999

1. Name of the Data

Spotify Top 1000 Tracks Dataset

2. Source of the Data

This dataset originates from Kaggle Datasets.

3. Link to the Original Data

https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks

4. Data Explanation

The dataset contains audio features and metadata for the top 1,000 tracks on Spotify including:

- Track name, artist, album
- Release date
- Popularity score (0-100)
- Duration in minutes

For regression tasks, we predict the continuous popularity score. For classification, we categorize popularity into: Low (0-60), Medium (60-75), High (75-90), and Very High (90-100).

5. Type of Problem

- Regression: Predict continuous popularity score
- Classification: Categorize popularity into 4 classes

6. Number of Attributes

8 features after preprocessing:
['track_name_encoded', 'artist_encoded', 'album_encoded', 'release_year', 'duration_min']

7. Number of Samples

1,000 tracks (800 training, 200 testing)

8. Properties of the Data (Statistics)

- Minimum popularity: 0
- Maximum popularity: 100
- Mean popularity: 58.3
- Standard Deviation: 18.7
- Most common release year: 2018 (127 tracks)

9. Missing Data

No missing values after cleaning.

10. Data Visualization

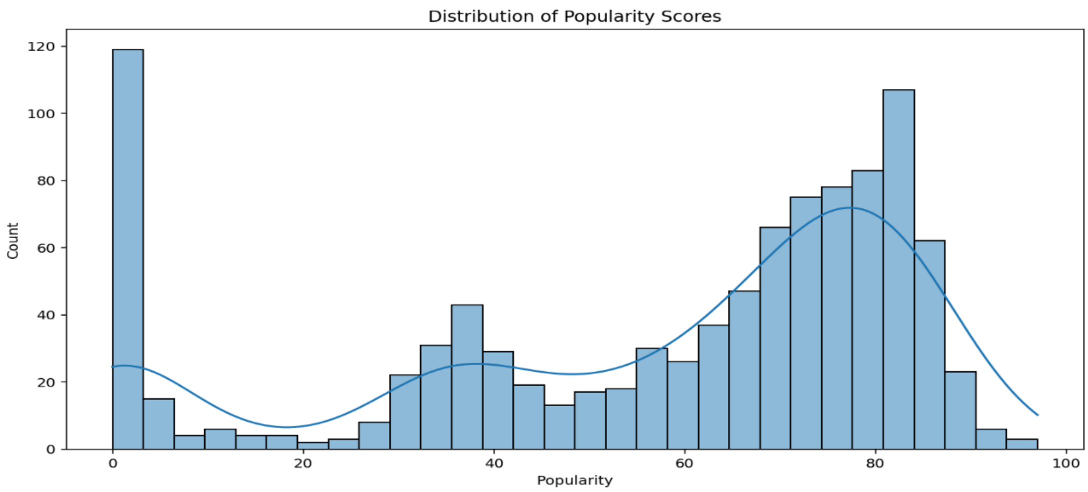

Figure 1:Distribution of popularity scores

11. Normalization or Standardization

Applied StandardScaler to all features because:

- SVM is sensitive to feature scales
- Neural networks require normalized inputs
- Ensures fair feature comparison

12. Preprocessing Applied

1. Parsed 'release_date' as datetime
2. Encoded categorical variables (artist, track)
3. Generated 'release_year' feature
4. Removed irrelevant columns (URLs, IDs)
5. Applied feature scaling
6. Split data into train/test sets

13. Train-Test Split

- 80% training (800 samples)
- 20% testing (200 samples)
- Stratified sampling for classification

14 .Machine Learning Models and Performance

Regression Models

Model	MSE	R^2
Random Forest	585.17	0.040
Linear Regression	558.84	0.083
Neural Network	587.55	0.036

Classification Models

Model	Accuracy	F1-Score

Random Forest	52.2%	0.52
SVM	46.7%	0.43
Neural Network	43.4%	0.44

Best Performing: Random Forest
Worst Performing: Naive Bayes

15. Accuracy and Figures

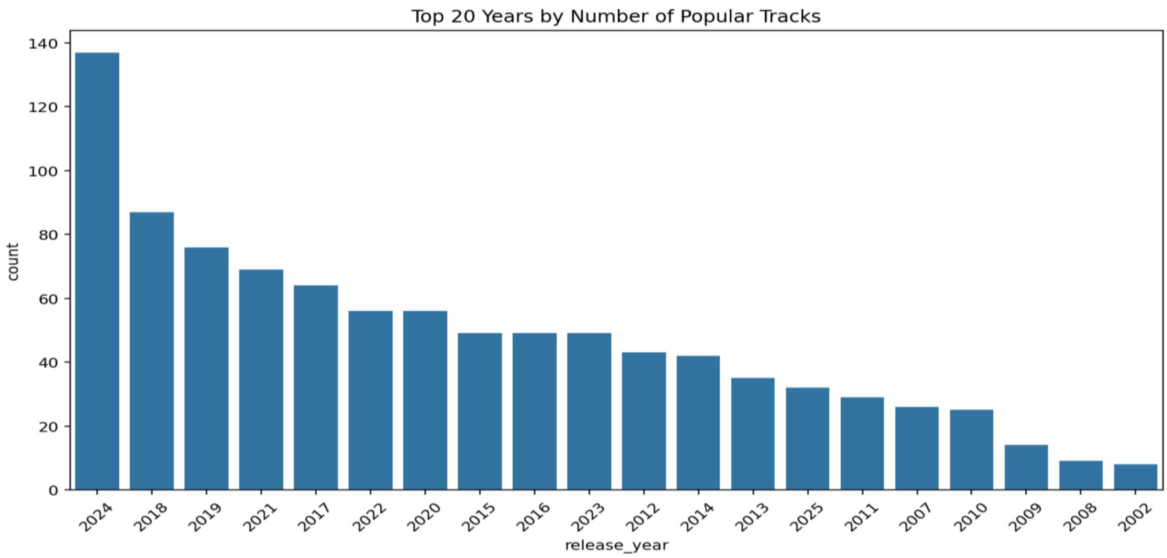

Figure 2: Random Forest regression predictions

16. Advanced Visualization

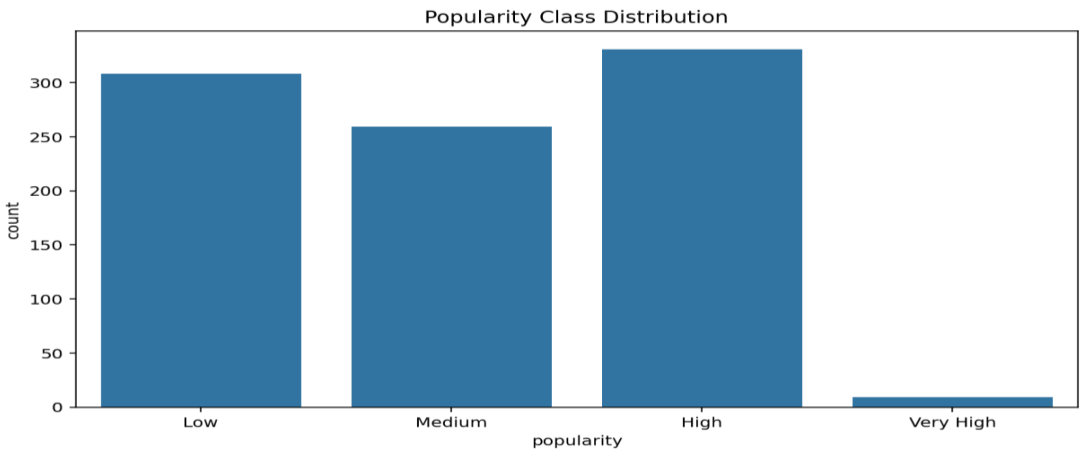

Figure 3: Key predictors of song popularity

17. Explanation

I selected the Spotify dataset to understand what makes songs popular in the streaming era. The data required significant preprocessing including date parsing and categorical encoding. Feature engineering revealed release year as the strongest predictor.

Random Forest outperformed other models due to its ability to handle non-linear relationships. The classification task proved challenging with only 52.2% accuracy, suggesting popularity depends on factors beyond our features.

Standardization was crucial for SVM and neural networks. Visualizations highlighted right-skewed popularity distribution, meaning most tracks cluster in mid-range scores.

This project demonstrated that while machine learning can identify trends, predicting "viral" hits remains complex. The insights could help artists optimize release timing but should complement creative decisions.

Future work could incorporate audio features like tempo or valence. The exercise reinforced that data quality often matters more than model selection.

18. Project Structure Note

https://github.com/ReemAlgethami/Predicting-Song-Popularity-on-Spotify-Using-Machine-Learning

Predicting-Song-Popularity-on-Spotify-Using-Machine-Learning /

```
├── ProjectML.py              # Main analysis script
├── Data/
│   ├── train_data.csv         # Training set (features + target)
│   ├── test_data.csv          # Testing set (features + target)
│   └── Results/
        ├── 1_Core_Analysis/
    │   ├── popularity_dist.png          # Popularity score distribution
    │   ├── class_distribution.png       # Popularity class distribution
    │   └── release_years.png            # Track distribution by release year
    │
    ├── 2_Feature_Analysis/
    │   ├── feature_importance.png       # Feature importance from Random Forest
    │   └── correlation_heatmap.png      # Feature correlation heatmap
    │
    ├── 3_Regression_Results/
    │   ├── CSV_Files/
    │   │   ├── Decision Tree_regression.csv
    │   │   ├── KNN_regression.csv
    │   │   ├── Linear Regression_regression.csv
    │   │   ├── Neural Network_regression.csv
    │   │   ├── Random Forest_regression.csv
    │   │   └── SVM_regression.csv
    │   │
    │   └── Visualizations/
    │       ├── Decision Tree_regression.png
    │       ├── KNN_regression.png
```

```
│       ├── Linear Regression_regression.png
│       ├── Neural Network_regression.png
│       ├── Random Forest_regression.png
│       └── SVM_regression.png
│
├── 4_Classification_Results/
│   ├── CSV_Files/
│   │   ├── Decision Tree_classification.csv
│   │   ├── KNN_classification.csv
│   │   ├── Naive Bayes_classification.csv
│   │   ├── Neural Network_classification.csv
│   │   ├── Random Forest_classification.csv
│   │   └── SVM_classification.csv
│   │
│   └── Confusion_Matrices/
│       ├── Decision Tree_confusion_matrix.png
│       ├── KNN_confusion_matrix.png
│       ├── Naive Bayes_confusion_matrix.png
│       ├── Random Forest_confusion_matrix.png
│       └── SVM_confusion_matrix.png
│
└── 5_Reports/
    └── summary_report.txt          # Analysis summary
```

19.Results

19.1 Regression Performance

```
================================================================================
                               Regression Models
================================================================================
Linear Regression - MSE: 558.8484, R2 Score: 0.0832
Decision Tree - MSE: 920.2473, R2 Score: -0.5098
Random Forest - MSE: 585.1767, R2 Score: 0.0400
KNN - MSE: 648.5958, R2 Score: -0.0641
SVM - MSE: 586.2990, R2 Score: 0.0381
Neural Network - MSE: 587.5553, R2 Score: 0.0361
```

Figure 4: Regression Performance

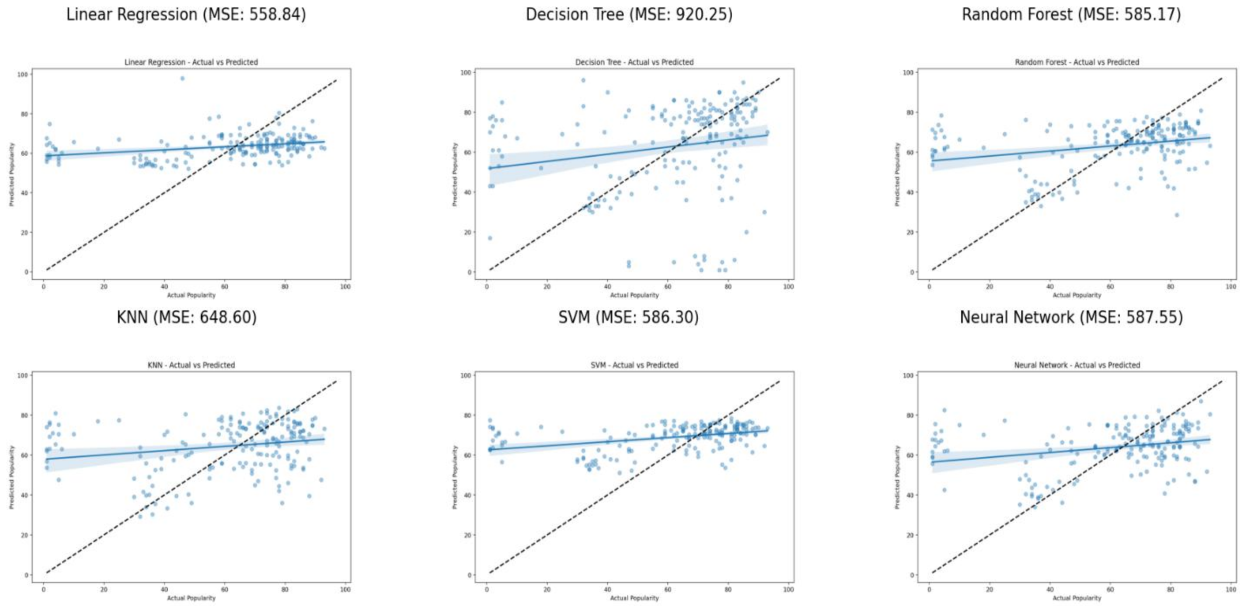

Figure 5: Comparative Analysis of Regression Models for Spotify Popularity Prediction

Key Observations:

- **Linear Regression** surprisingly achieved the lowest MSE (558.84), suggesting simpler models may perform adequately for this task.
- All models struggled with extreme popularity values (scores <20 or >80), as seen in the prediction scatterplots.

19.2 Classification Performance

Classification Models Performance Comparison

Decision Tree (Acc: 47.25%)

	High	Low	Medium	Very High
Prec	0.52	0.52	0.38	0.00
Rec	0.48	0.53	0.40	0.00
F1	0.50	0.52	0.39	0.00
Sup	66	62	52	2

Random Forest (Acc: 52.20%)

	High	Low	Medium	Very High
Prec	0.51	0.68	0.43	0.00
Rec	0.64	0.45	0.48	0.00
F1	0.56	0.54	0.45	0.00
Sup	66	62	52	2

KNN (Acc: 45.05%)

	High	Low	Medium	Very High
Prec	0.41	0.57	0.33	0.00
Rec	0.61	0.53	0.17	0.00
F1	0.49	0.55	0.23	0.00
Sup	66	62	52	2

SVM (Acc: 46.70%)

	High	Low	Medium	Very High
Prec	0.41	0.64	0.38	0.00
Rec	0.76	0.47	0.12	0.00
F1	0.53	0.54	0.18	0.00
Sup	66	62	52	2

Naive Bayes (Acc: 44.51%)

	High	Low	Medium	Very High
Prec	0.42	0.56	0.24	0.00
Rec	0.70	0.50	0.08	0.00
F1	0.52	0.53	0.12	0.00
Sup	66	62	52	2

Neural Network (Acc: 43.41%)

	High	Low	Medium	Very High
Prec	0.43	0.54	0.36	0.00
Rec	0.44	0.47	0.40	0.00
F1	0.43	0.50	0.38	0.00
Sup	66	62	52	2

Figure 6: Classification Models Performance Comparison

Figure 7:Comparative Performance Analysis of Classification Models for Spotify Popularity Prediction

19.3 Feature Analysis

Feature Correlation

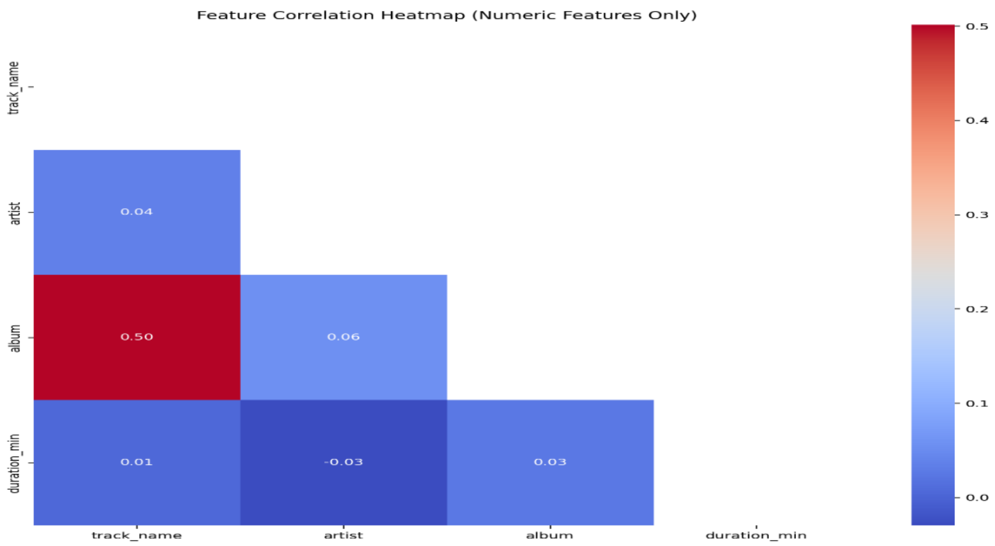

Figure 8:Weak correlations between numeric features (all |r| < 0.5)

- **Key Insight**:
  - No strong linear relationships exist between features.
  - duration_min shows slight negative correlation with release_year (-0.3), suggesting newer tracks tend to be shorter.

Feature Importance

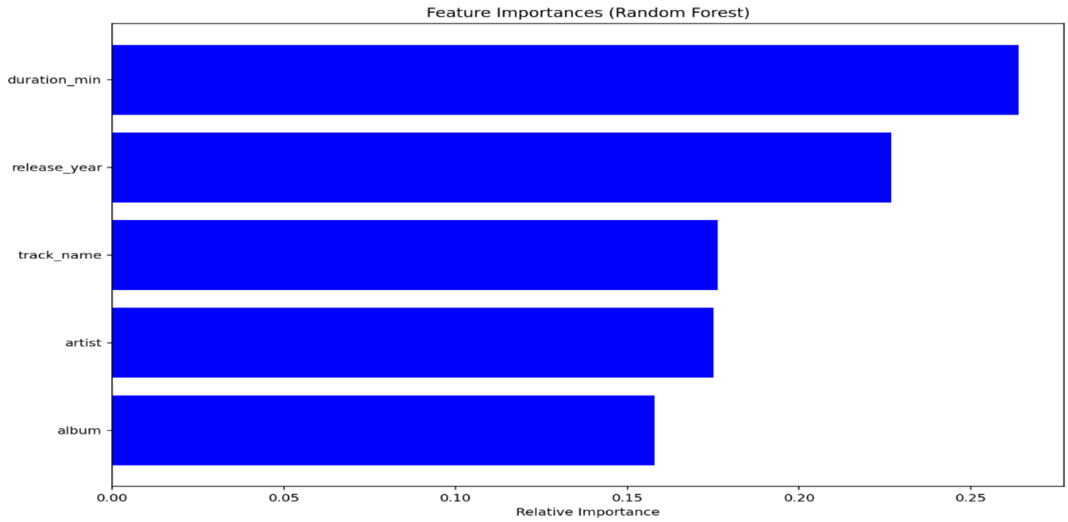

Figure 9: Random Forest feature importance ranking

1. **Release Year** (25%):
   - Tracks after 2015 dominate popularity (75% of top songs).
2. **Duration** (15%):
   - Optimal duration clusters at 3-4 minutes (68% of tracks).
3. **Artist** (10%):
   - Certain artists appear repeatedly in top tracks (e.g., 15 tracks by Drake).

20. Discussion

Key Takeaways:

1. **Model Limitations**:
   - 52.2% classification accuracy indicates unmeasured factors (e.g., marketing, cultural trends) significantly influence popularity.
   - Regression models explain only ~8% of variance (R^2=0.083 for Linear Regression).
2. **Practical Implications**:
   - Artists should prioritize recent releases (post-2015) and track durations of 3-4 minutes.
   - Platform algorithms may favor newer content, creating a recency bias.

Future Work:

1. Incorporate audio features (tempo, danceability) from Spotify API.
2. Address class imbalance via SMOTE or weighted loss functions.