SPOTIFY

OUR TEAM

- Rifqi Khalis
- Nikhil Gaikwad
- Robert Chan
- Abdul Hadi

What we'll be covering?

1. Introduction
2. Challenges
3. Reconstruction Schematic
4. Algorithm
5. Logistic Regression
6. KNN
7. Decision Tree
8. Recommending the Next Song

Spotify

Introduction

Music streaming services like Spotify have revolutionized how we enjoy music, offering personalized playlists and shared family subscriptions.

However, on November 21, Spotify's servers faced a major hacker attack that deleted users' Top-of-the-Year playlists from family accounts.

This resulted in one mixed playlist for each account, blending the diverse tastes of all family members.

Challenges

1. To reconstruct each user's Top-of-the-Year playlists for each year (from 2018 to 2024).

2. To propose a way to construct such recommendation algorithms and to suggest several features of the song that can be used as inputs for the algorithm

Challenges

Reconstruction Schematic

1. Understanding the data
2. Preprocessing Data
3. Build Algorithm Model
4. Train and test model using known data
5. Evaluate the metrics
6. Choose and use the best model based on metrics to predict missing data
7. Save and export to CSV file based on reconstructed Top-of-the-Year playlist of a specific user at a specific year

RECONSTRUCTION

Build Algorithm Model for Reconstruction

Targets

User

Logistic Regression

Top_year

1. K-nearest neighbor (KNN)
2. Decision Tree

ALGORITHM

User Reconstruction using Logistic Regression

Feature Engineering

- Count user-artist interactions.
- Merge features into the dataset.
- Removed 'unknown' user entries to improve data quality.

↓

Choose Relevant Features

- Dropped irrelevant columns (e.g., name, album)
- Encoded categorical variables
- Encoded target variable (user) using LabelEncoder

↓

Create Pipeline

- Polynomial Features (Degree: 1)
- Standard Scaler
- Logistic Regression with OneVsRest and L2 Penalty

↓

Train & Test the model

- Split Data: Train (70%) / Test (30%) — Random State = 1.
- Evaluated metrics and performance.

User Reconstruction using Logistic Regression

Training Data Classification Report:

	precision	recall	f1-score	support
0	0.85	0.80	0.83	476
1	0.85	0.85	0.85	483
2	0.86	0.86	0.86	478
3	0.88	0.94	0.91	507
4	0.91	0.91	0.91	498
accuracy			0.87	2442
macro avg	0.87	0.87	0.87	2442
weighted avg	0.87	0.87	0.87	2442

Test Data Classification Report:

	precision	recall	f1-score	support
0	0.84	0.77	0.81	224
1	0.89	0.85	0.87	217
2	0.86	0.85	0.85	218
3	0.81	0.96	0.88	188
4	0.90	0.88	0.89	200
accuracy			0.86	1047
macro avg	0.86	0.86	0.86	1047
weighted avg	0.86	0.86	0.86	1047

Top Year Reconstruction using KNN

Feature Engineering

- Counted album appearances as a top song in a given year.
- Merged the count matrix back into the dataset.
- Removed 'unknown' entries to improve data quality.

↓

Choose Relevant Features

- Dropped irrelevant columns (user, top_year, etc.).
- Encoded categorical variables for numerical compatibility.

↓

Create Pipeline

- Standard Scaler for feature normalization.
- KNN Classifier: k = 7 (matches 7 classes: 2018–2024).
- Tuned hyperparameters with GridSearchCV:
  - Weights: uniform, distance
  - Metrics: euclidean, manhattan.

↓

Train & Test the model

- Split Data: Train (70%) / Test (30%).
- Used cross-validation (cv=10) to evaluate accuracy.

Top Year Reconstruction using KNN

```
Best Parameters: {'knn__metric': 'manhattan', 'knn__n_neighbors': 7, 'knn__weights': 'uniform'}
```

Training Data Classification Report:

	precision	recall	f1-score	support
0	0.80	0.88	0.83	337
1	0.79	0.80	0.79	348
2	0.79	0.78	0.78	347
3	0.74	0.81	0.77	343
4	0.74	0.70	0.72	370
5	0.73	0.69	0.71	348
6	0.82	0.75	0.78	349
accuracy			0.77	2442
macro avg	0.77	0.77	0.77	2442
weighted avg	0.77	0.77	0.77	2442

Test Data Classification Report:

	precision	recall	f1-score	support
0	0.73	0.80	0.76	163
1	0.73	0.74	0.73	150
2	0.66	0.71	0.68	148
3	0.65	0.69	0.67	153
4	0.49	0.54	0.51	130
5	0.58	0.47	0.52	152
6	0.74	0.63	0.68	151
accuracy			0.66	1047
macro avg	0.65	0.65	0.65	1047
weighted avg	0.66	0.66	0.66	1047

Top Year Reconstruction using Decision Tree

Feature Engineering

- Prepared features for predicting the top_year
- Removed irrelevant columns and encoded variables

Hyperparameter Search

- Used RandomizedSearchCV for hyperparameter tuning.
- Defined search space:
  - Criterion: gini, entropy
  - Max Depth: Random range (10–50)
  - Min Samples Split: Random range (2–30)
  - Min Samples Leaf: Random range (1–30)
  - Max Features: sqrt, log2, None
  - Min Impurity Decrease: Uniform (0–0.01).

Cross-Validation

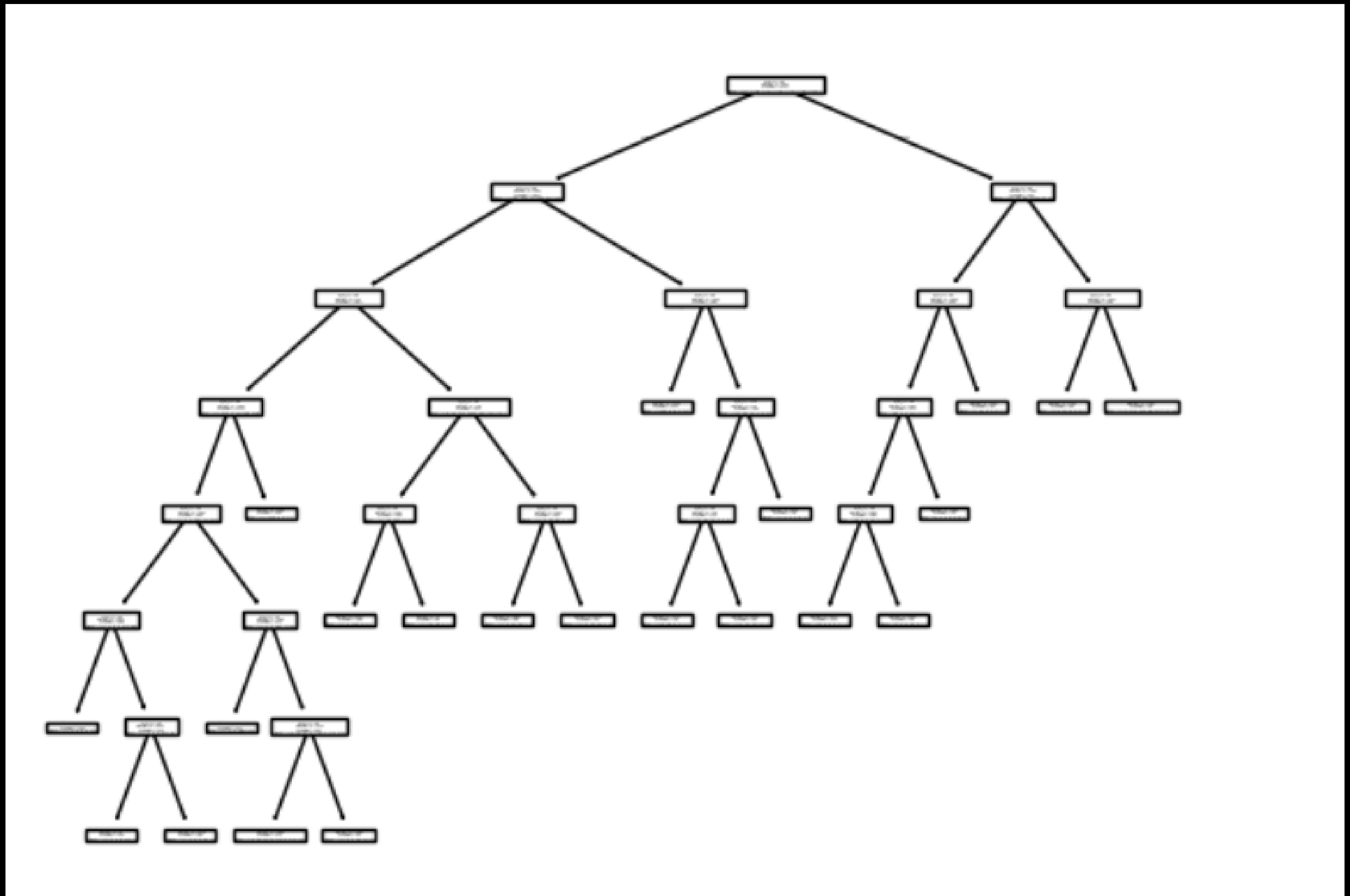
- Performed 10-fold cross-validation for model evaluation.
- Sampled 20 iterations for hyperparameter search.

Train & Test the model

- Scoring Metric: Accuracy.
- Random State = 0 (Reproducibility).
- Best model automatically refitted after tuning.

Top Year Reconstruction using Decision Tree

```
Best Parameters: {'criterion': 'entropy', 'max_depth': 46, 'max_features': None, 'min_impurity_decrease': 0.009719450024996659, 'min_samples_leaf': 4,
'min_samples_split': 4}
```



RECONSTRUC
TION

Top Year Reconstruction using Decision Tree

Training Data Classification Report:

	precision	recall	f1-score	support
0	0.86	0.83	0.85	337
1	0.68	0.94	0.79	348
2	0.80	0.73	0.77	347
3	0.87	0.80	0.84	343
4	0.73	0.81	0.77	370
5	0.75	0.72	0.73	348
6	0.99	0.72	0.83	349
accuracy			0.79	2442
macro avg	0.81	0.79	0.79	2442
weighted avg	0.81	0.79	0.79	2442

Test Data Classification Report:

	precision	recall	f1-score	support
0	0.84	0.78	0.81	163
1	0.64	0.91	0.75	150
2	0.78	0.76	0.77	148
3	0.87	0.79	0.83	153
4	0.63	0.78	0.70	130
5	0.71	0.62	0.66	152
6	0.97	0.69	0.81	151
accuracy			0.76	1047
macro avg	0.78	0.76	0.76	1047
weighted avg	0.78	0.76	0.76	1047

RECONSTRUCTION

Recommending the Next Song

A Spotify Algorithmic Approach

Framework for Song Recommendation

Step 1	Identify user preferences and contextual factors.
Step 2	Leverage features of the current song and historical listening behavior.
Step 3	Develop a hybrid model combining content-based and collaborative filtering.
Step 4	Factor in additional elements like mood, seasonality, and events (e.g., festivals).

What Data Do We Need?

Song Features:

Current song attributes (tempo, key, loudness, etc.).

Mood and sentiment (e.g., happy, sad, energetic).

Popularity metrics (global and seasonal trends).

Contextual Data:

Year and month.

Festivals, holidays, or special occasions.

User Behavior:

Listening history (genre, tempo, artist preferences).

Skip rate or song completion rate.

Recommendation System

User-base filtering

User 1 has rated some songs (binary rating = like a song)

User/Song	Song A	Song B	Song C	Song D	Song E
User 1	1	0	?	1	0
User 2	1	?	1	?	1
User 3	?	1	1	?	0
User 4	1	1	1	0	?
User 5	?	0	1	1	1

$$cosine\ \ similarity\ (A, B) = \frac{A\ \cdot B}{||A||\ \ ||B||}$$

User 2 is the most similar to user 1

User	Song A	Song B	Song C	Song D	Song E
User 1	1.0000	0.8165	0.0000	0.7070	0.5000
User 2	0.8165	1.0000	0.5000	0.7070	0.5000
User 3	0.0000	0.5000	1.0000	0.0000	0.7070
User 4	0.7070	0.7070	0.0000	1.0000	0.5000
User 5	0.5000	0.5000	0.7070	0.5000	1.0000

Content-based filtering

User 1 has listen to song A

Song	Valence	Tempo (BPM)	Energy
Song A	0.90	120	0.70
Song B	0.85	115	0.75
Song C	0.40	95	0.30
Song D	0.20	70	0.40

$$cosine\ \ similarity\ (A, B) = \frac{A\ \cdot B}{||A||\ \ ||B||}$$

Song Pair	Song A	Song B	Song C	Song D
Song A	1.0000	0.9999	0.2380	0.0720
Song B	0.9999	1.0000	0.2190	0.0730
Song C	0.2380	0.2190	1.0000	0.6210
Song D	0.0720	0.0730	0.6210	1.0000

song	rank	cosine
B	1	0.999957
C	2	0.238014
D	3	0.071783

Hybrid system

User base:
Look who give similar rating with the user and give a list of song

Song	Content-Based Score	Collaborative Filtering Score	Final Score
Song A	0.8	0.7	0.76
Song B	0.6	0.5	0.56
Song C	0.9	0.8	0.86
Song D	0.7	0.6	0.66

Content-based:
Look which song are similar to the one the user like and give a list of song

Give song recomendation

Larana Inc.

THANK YOU