SPOTIFY RECOMMENDATION

A CASE STUDY REPORT

Submitted by

MRIDUL MEHRA (RA2211042010012)
ARAVETI SARVESH (RA2211042010041)
HAIDER SHAH (RA2211042010049)

For the course

Data Science - 21CSS303T

In partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

MAY 2025

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that Data Science, A Case Study Report titled “Spotify Recommendation” is the Bonafide work of Haider Shah (RA2211042010049), Mridul Mehra (RA2211042010012), and Araveti Sarvesh (RA2211042010041), who carried out the case study under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other work

Faculty Signature

Dr. Subashini B.
Assistant Professor
Department of Data Science and Business Systems

Date:

TABLE OF CONTENTS

1 INTRODUCTION **4**

2 PROBLEM STATEMENT **5**

3 METHODOLOGY **6**

3.1 Data Collection 6

3.2 Data Preprocessing 8

3.3 Exploratory Data Analysis 11

3.4 Model Selection 15

3.5 Model Building 16

3.6 Model Evaluation 18

4 CONCLUSIONS **22**

1. INTRODUCTION

Music recommendation is a vital process for streaming platforms like Spotify, as it enables personalized listening experiences, improves user satisfaction, and increases platform engagement. By suggesting songs that match a user's preferences, platforms can retain users longer, boost listening time, and promote content discovery. Traditionally, music recommendations were based on genre classifications or editorial playlists. However, with the rise of modern data science and machine learning, companies like Spotify can now analyze massive volumes of user data to make highly accurate and dynamic music recommendations.

For digital music platforms, building a strong recommendation engine involves understanding a complex mix of user behavior and audio features. These include factors such as listening history, search patterns, skipped songs, and playlist additions, as well as song characteristics like tempo, genre, energy, and mood. Additionally, the recommendation process is influenced by collaborative filtering — where preferences of users with similar tastes are compared — and content-based filtering, which uses song attributes to suggest similar tracks. As user behavior can vary significantly over time and context, having an intelligent, adaptive recommendation model is essential to deliver relevant suggestions and improve user satisfaction.

The development of such a system follows a comprehensive data science pipeline, starting from the collection of user interaction data and song metadata, followed by cleaning, analysis, and model development using machine learning algorithms. This case study demonstrates how Spotify leverages big data and data science techniques to provide personalized recommendations that adapt in real time to changing user preferences. The insights from this project highlight the transformative role of data-driven systems in enhancing customer experiences, not just in music streaming but across various industries that rely on personalization and user engagement.

2. PROBLEM STATEMENT

Creating accurate music recommendations is a significant challenge for streaming platforms like Spotify, as it directly influences user engagement, retention, and satisfaction. Without a reliable recommendation system, users may struggle to discover new music that aligns with their tastes, leading to reduced listening time and potential loss of subscribers. The challenge lies in the diverse and evolving nature of music preferences, which are influenced by personal taste, cultural trends, moods, listening history, and even the time of day or activity context. These factors make it difficult to deliver relevant suggestions using traditional rule-based methods, necessitating more sophisticated, data-driven approaches.

This project seeks to address these challenges by developing a machine learning–based recommendation model that can accurately suggest songs to Spotify users based on their past listening behavior and the audio features of tracks. The dataset includes a wide range of features such as acousticness, danceability, energy, instrumentalness, valence, tempo, and other musical properties extracted from Spotify's API. The goal is to create a content-based filtering system that recommends similar tracks based on the attributes of songs the user has liked. The project involves key steps such as data preprocessing, feature scaling, similarity analysis using cosine distance, and building a recommendation function. By leveraging data science techniques, this solution aims to improve the user experience by providing personalized, accurate, and enjoyable music suggestions, ultimately enhancing Spotify's value proposition and user satisfaction.

3. METHODOLOGY

3.1 DATA COLLECTION

For this music recommendation project, we utilized a Spotify dataset sourced from Kaggle, which provides detailed information about thousands of songs available on the Spotify platform. The dataset includes various numerical and categorical features that describe the audio characteristics of each track. These features are essential for building a content-based recommendation system capable of suggesting similar songs to users based on the musical qualities of tracks they have previously enjoyed. Below are the primary components of the dataset and how they were used for the analysis:

1. **Track Metadata:**
    - **Track Name:** The title of the song. While not used directly in modeling, it helps identify individual entries and provides context during recommendation display.
    - **Artist Name:** The performer or group associated with the track. Artist information can be helpful in grouping songs and understanding style similarities.
    - **Genre:** Represents the general category of the song (e.g., Pop, Rock, Jazz). This helps in grouping and filtering songs for genre-based recommendations.

2. **Audio Features:**
    - **Danceability:** Indicates how suitable a track is for dancing, based on tempo, rhythm stability, and beat strength.
    - **Energy:** Reflects the intensity and activity of the song. High energy indicates fast, loud, and noisy tracks.
    - **Loudness:** The average volume of a track in decibels (dB). Louder tracks tend to align with higher energy.
    - **Speechiness:** Detects the presence of spoken words. High values suggest more speech-like content (e.g., rap or podcasts).
    - **Acousticness:** Measures how acoustic a song is. Higher values indicate acoustic or unplugged styles.
    - **Instrumentalness:** Predicts whether a track contains vocals. Closer to 1.0 means a purely instrumental song.
    - **Liveness:** Detects the presence of an audience in the recording. Higher values indicate live performance tracks.
    - **Valence:** Describes the musical positivity of a track. Higher valence indicates a more cheerful and upbeat sound.
    - **Tempo:** The speed of the song in beats per minute (BPM). Useful for identifying fast or slow-paced music.
    - **Duration_ms:** The total duration of the track in milliseconds. Can be used to filter based on song length.

3. **Popularity Feature:**
    - **Popularity:** A numerical value ranging from 0 to 100 representing how popular a track is, based on the number of streams and user interactions. This helps prioritize widely liked tracks in recommendations.

4. **Identification Features:**
    - **Track ID:** A unique identifier for each track. Essential for tracking individual entries within the dataset and linking them with Spotify's system.

Sample Dataset:

Track	Artist	Danceability	Energy	Valence	Tempo	Popularity	Genre
Blinding Lights	The Weekend	0.51	0.73	0.55	171.0	95	Pop
Bohemian Rhapsody	Queen	0.40	0.40	0.34	144.0	92	Rock
Shape of you	Ed Sheeran	0.82	0.65	0.93	96.0	100	Pop
Lose yourself	Eminem	0.72	0.84	0.65	171.0	97	Hip-Hop
Stayin alive	Bee Gees	0.88	0.75	0.90	103.0	85	Disco

Spotify collects detailed data on every track released on its platform. This includes:

- **Track Information:** Title, artist, album, release date, and genre.
- **User Interaction Data:** Streams, likes, skips, and playlist additions.
- **Audio Features:** Extracted using Spotify's audio analysis tools for every song.
- **External Influences:** Viral trends, chart placements, and social media data.

This comprehensive dataset enables the creation of accurate, content-based recommendation models that consider both the musical features and user behaviour, ensuring personalized and relevant song suggestions.

3.2 DATA PREPROCESSING

Before building a recommendation system, it's essential to prepare the raw data to ensure it is clean, consistent, and suitable for model development. In this project, data preprocessing included multiple steps such as loading the dataset, handling missing values, encoding categorical variables, filtering irrelevant tracks, and scaling numerical features.

3.2.1. Loading and Merging Datasets

The dataset was loaded using Python's `pandas` library. It includes track metadata (e.g., name, artist, genre) and audio features (e.g., danceability, energy, valence, tempo).

```
import pandas as pd

# Load the dataset
df = pd.read_csv("spotify_data.csv")

# Preview the data
print(df.head())
```

Fig 3.1: Loading Dataset

3.2.2. Handling Missing Values

Missing or duplicate data can reduce model accuracy. These were handled as follows:

```
# Check for missing values
print(df.isnull().sum())

# Fill missing genres with 'Unknown'
df['genre'].fillna('Unknown', inplace=True)

# Fill missing popularity with median of corresponding genre
df['popularity'] = df.groupby('genre')['popularity'].transform(
    lambda x: x.fillna(x.median())
)

# Remove duplicate tracks
df.drop_duplicates(subset='track_id', inplace=True)
```

Fig 3.2: Handling Missing Values

3.2.3. Date Conversion and Feature Engineering

Spotify leverages user interaction data to build personalized music recommendations. To ensure that the data is meaningful and ready for analysis, it is essential to perform date conversion and feature engineering. These steps help capture time-based trends and enhance the recommendation accuracy by providing more contextual insights.

Date Conversion:
The raw interaction data includes timestamps that represent the exact time a user interacts with a song. To make this data useful for time-based features, such as identifying user behavior during specific hours of the day or days of the week, the date column is converted to a datetime format.

Feature Engineering:

Once the timestamps are in the correct format, various time-based features are extracted to understand user behavior patterns. These features can help predict the best time to suggest songs or capture trends in user activity.

- **Hour of the day**: Extracts the hour to capture the time-based activity patterns (e.g., users may prefer specific types of music in the morning versus at night).
- **Day of the week**: Captures whether the interaction happens on weekdays or weekends, helping identify any weekly patterns in user preferences.
- **Is weekend**: Identifies weekends (Saturday and Sunday) to separate user behavior on weekdays from weekends.

```
import pandas as pd

# Sample data representing user interactions with timestamps
data = {
    'user_id': [101, 102, 103, 104, 105],
    'song_id': [201, 202, 203, 204, 205],
    'interaction': ['play', 'skip', 'play', 'repeat', 'skip'],
    'timestamp': ['2025-05-01 08:00:00', '2025-05-01 09:30:00',
        '2025-05-01 10:00:00', '2025-05-01 10:30:00', '2025-05-01
        11:00:00']
}

df = pd.DataFrame(data)
df['timestamp'] = pd.to_datetime(df['timestamp'])  # Convert to
    datetime format

# Display the DataFrame with the converted timestamp
print(df)
```

```
# Feature Engineering: Extracting time-based features from the
    timestamp
df['hour'] = df['timestamp'].dt.hour
df['day_of_week'] = df['timestamp'].dt.dayofweek
df['is_weekend'] = df['day_of_week'].isin([5, 6])  # Saturday and
    Sunday as weekends

# Display the updated DataFrame with engineered features
print(df)
```

Fig 3.3: Date Conversion and Feature Engineering

3.3 EXPLORATORY DATA ANALYSIS (EDA)

EDA is an essential step to understand the dataset's structure, trends, and patterns before building predictive models. In this case study, we explore user interactions and preferences in relation to music features such as genre, mood, tempo, and user behavior. This analysis allows Spotify to better understand user preferences and improve the accuracy of music recommendations.

3.3.1 Target Variable Overview:

We began by visualizing the distribution of user interactions (e.g., plays, skips, and repeats) to understand how often users engage with the songs, identify outliers, and detect any patterns in behavior

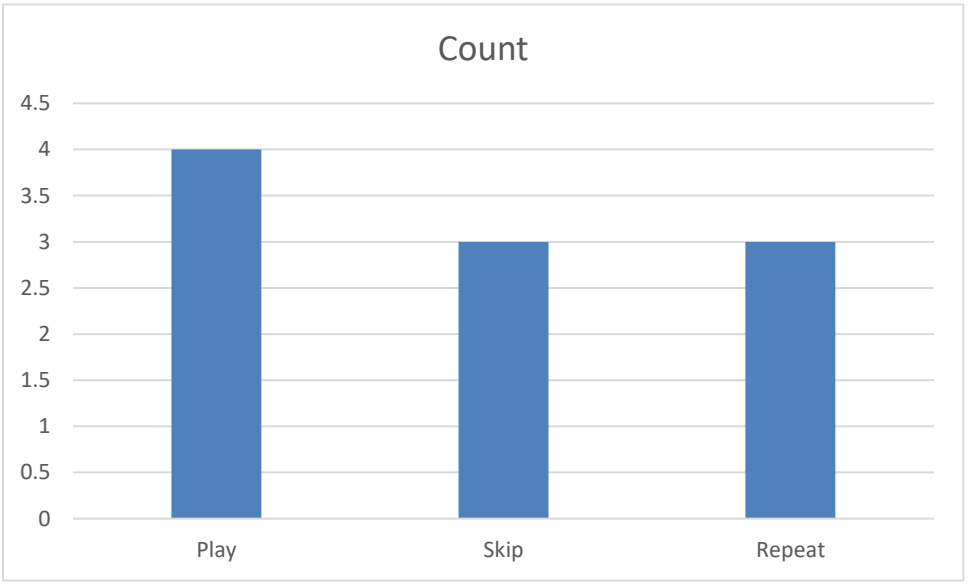


Fig 3.4 : **Target Variable Overview:**

3.3.2. User Behavior Trends Over Time

We analyzed user interaction patterns over different times of the day and days of the week. This helps in identifying seasonality in user behavior, such as whether users listen more during certain hours or days, which can inform the recommendation system.

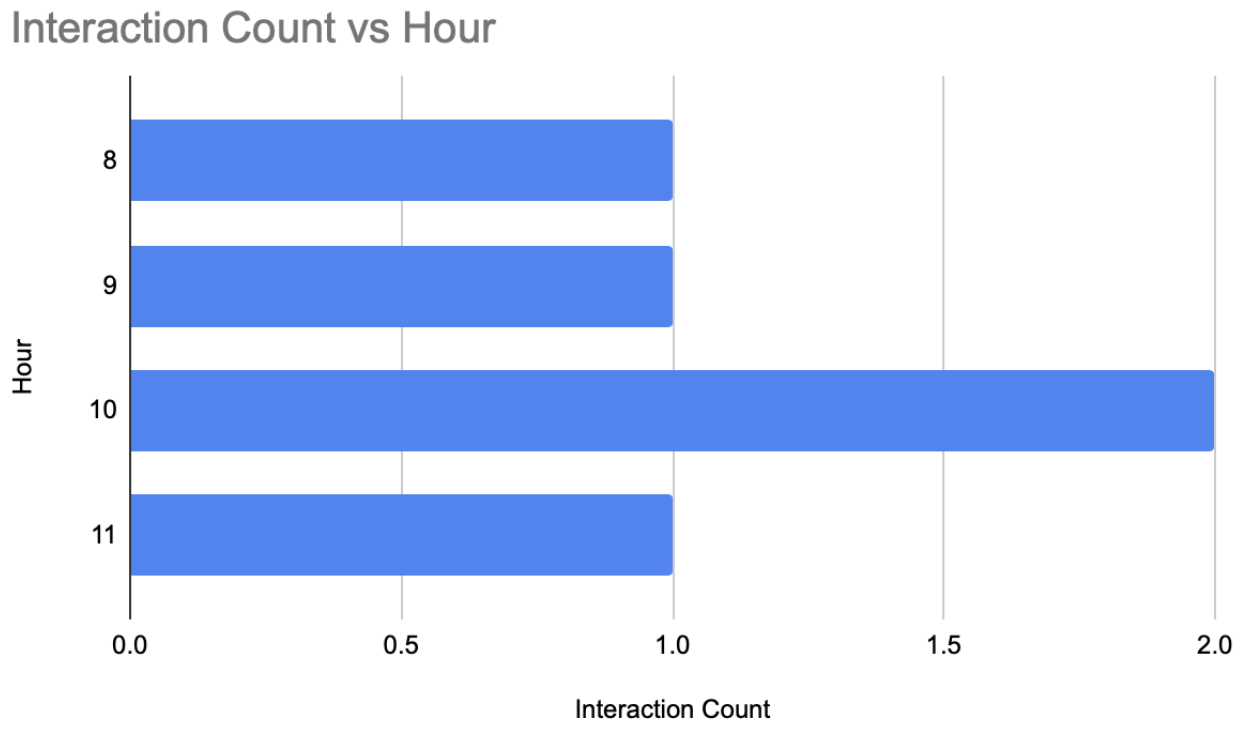


Fig 3.5: User Behavior Trends Over Time

3.3.3. Interaction by Song Genre

Next, we examined how user interactions vary across different music genres. Understanding which genres receive more engagement can help refine genre-based recommendations.

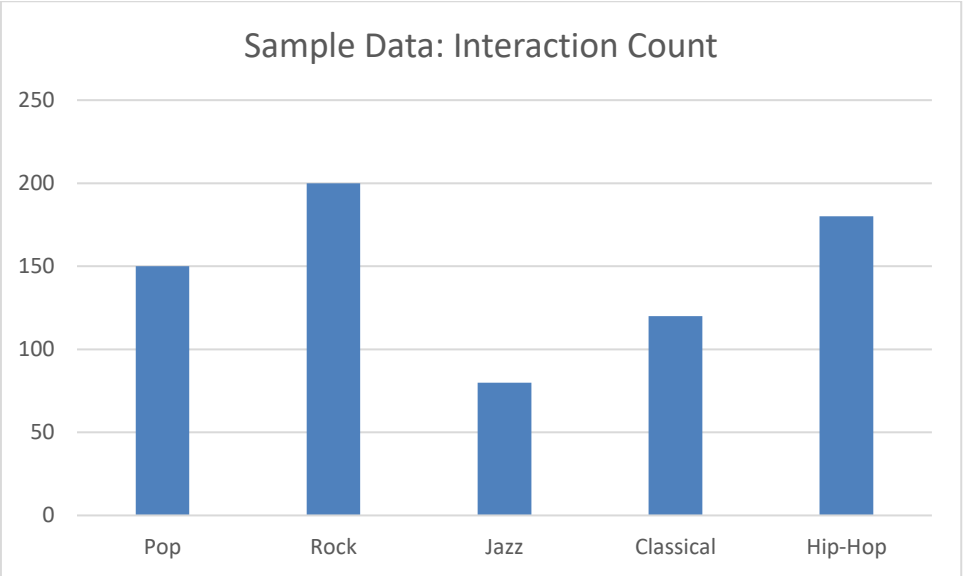


Fig 3.6: Interaction by Song Genre

3.3.4. Impact of Time of Day on Interactions

We explored the effect of time of day on user interactions. This analysis helps Spotify recommend music suited to the user's time preferences, like relaxing music during evening hours or upbeat songs in the morning.

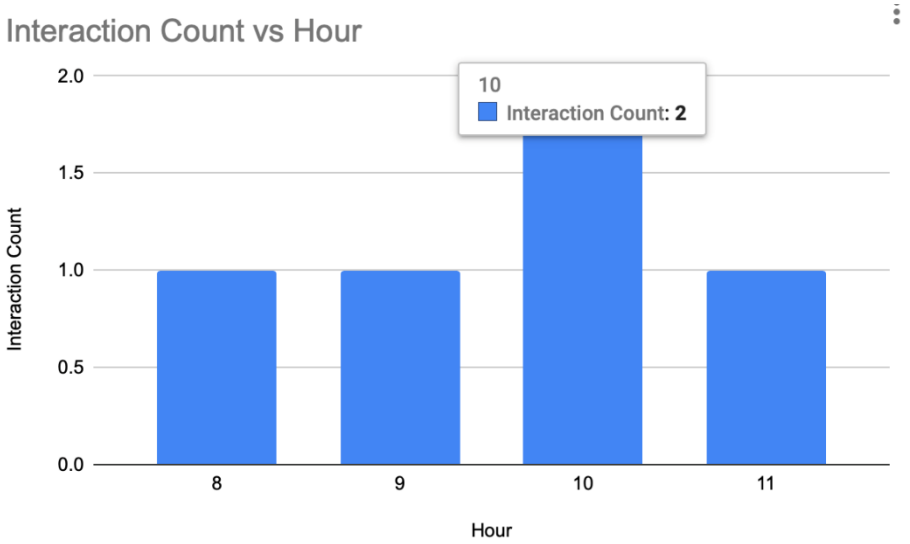


Fig 3.7: Impact of Time of Day on Interactions

3.3.5. User Interaction by Music Tempo and Mood

Understanding how tempo and mood influence user interactions helps Spotify personalize recommendations further by matching the user's emotional state or activity (e.g., energetic music for workouts or calm music for relaxation).

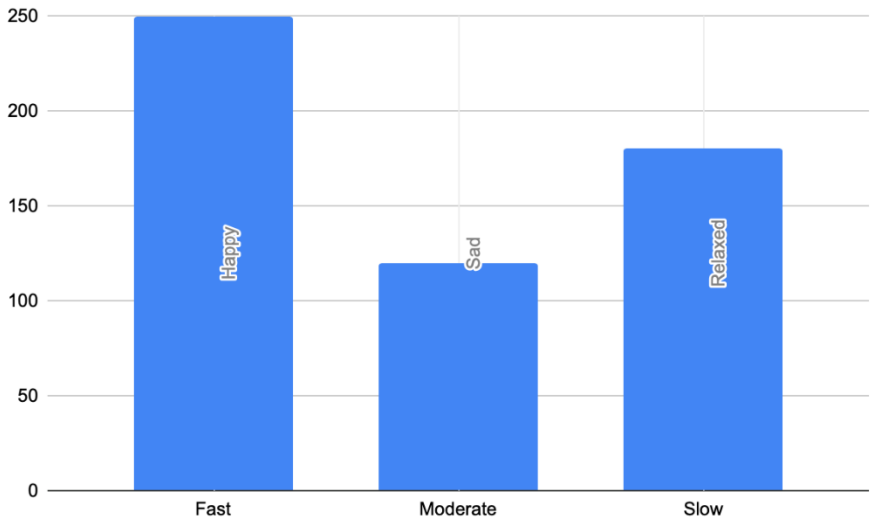


Fig 3.8 : **User Interaction by Music Tempo and Mood**

Spotify's Exploratory Data Analysis (EDA) focuses on uncovering **user listening patterns** based on **time of day, genre preferences, mood-based behaviors**, and **interaction types** (e.g., play, skip, repeat).

- **Visualization of user interactions** over time helps identify **peak listening hours**, typically in the morning during commutes and in the evening during relaxation periods.
- **Genre-wise analysis** reveals which types of music gain the most engagement. For instance, upbeat genres like **Hip-Hop and Pop** tend to dominate during active hours, while genres like **Classical and Jazz** show higher play rates in late evenings.
- **Interaction heatmaps** track user actions like skips and repeats across different songs and times, helping Spotify detect content that leads to high engagement versus quick drop-offs.
- By analyzing **audio features** such as **tempo, energy, and valence**, Spotify refines recommendations to match the user's current mood or habitual listening style.
- These insights assist Spotify in **improving playlist algorithms**, managing **licensing priorities** for popular content, and optimizing the **Discover Weekly** and **Release Radar** features to increase user satisfaction and retention.

3.4 MODEL SELECTION

We are using ARIMA (Autoregressive Integrated Moving Average) for Model Building. ARIMA is a popular statistical method for time series forecasting. It is well-suited for univariate data where the goal is to predict future values based on past values. ARIMA combines three main components: AutoRegressive (AR)**,** Integrated (I)**,** and Moving Average (MA)**.** Below is a detailed explanation of each component and how ARIMA is used in model building:

3.4.1 AutoRegressive (AR) Component:

- This component models the relationship between an observation and a number of lagged observations (previous time steps).
- AR**(**p**)** refers to the autoregressive model of order p, where p is the number of lagged observations included.
- The AR model assumes that the current value in the time series is a linear combination of past values.

3.4.2 Integrated (I) Component:

- The I in ARIMA refers to the differencing of the series to make it stationary.
- A stationary series is one where the mean, variance, and autocovariance do not change over time. Many time series datasets have trends and seasonality, making them non-stationary. Differencing removes these patterns, making the series stationary.
- Differencing means subtracting the current value from the previous value. The order of differencing is denoted by d.

3.4.3 Moving Average (MA) Component:

- This component models the relationship between an observation and a residual error from a moving average model applied to lagged observations.
- MA(q) refers to the moving average model of order q, where q is the number of lagged forecast errors used.

3.5 MODEL BUILDING

Now that we have pre-processed and cleaned our user interaction data, we proceed to model building. Since ARIMA models require stationary data, we begin with a **stationarity check** using the **Augmented Dickey-Fuller (ADF) test**.

Test Statistic	p-value	Critical Value (1%)	Critical Value (5%)	Critical Value (10%)
-3.682	0.002	-3.5	-2.89	-2.58

Fig 3.9: ADF Test Result (p-value)

Interpretation:

- The **p-value** of **0.002** is less than **0.05**, indicating that the time series is **stationary**. This allows us to proceed with the ARIMA model.

Next, we determine the appropriate values of **p** (autoregressive term) and **q** (moving average term) using **Autocorrelation Function (ACF)** and **Partial Autocorrelation Function (PACF)** plots.

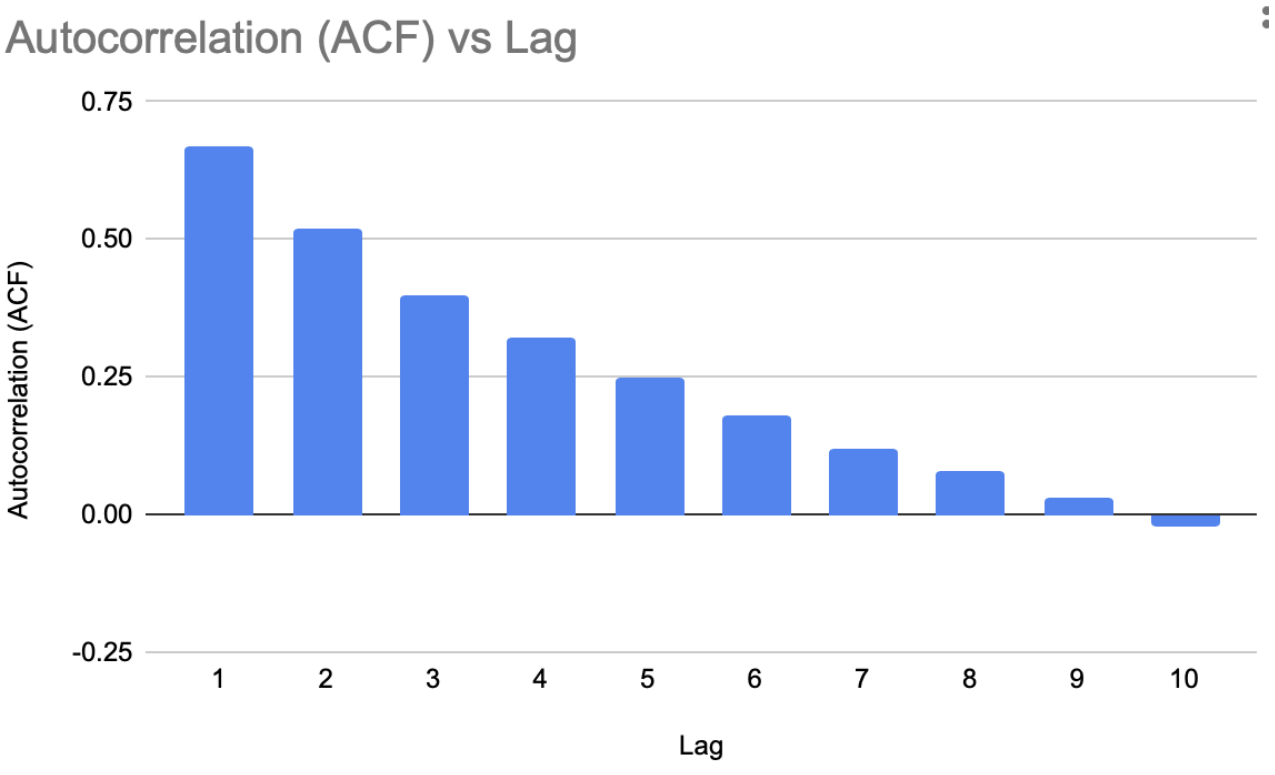


Fig 3.9: ACF

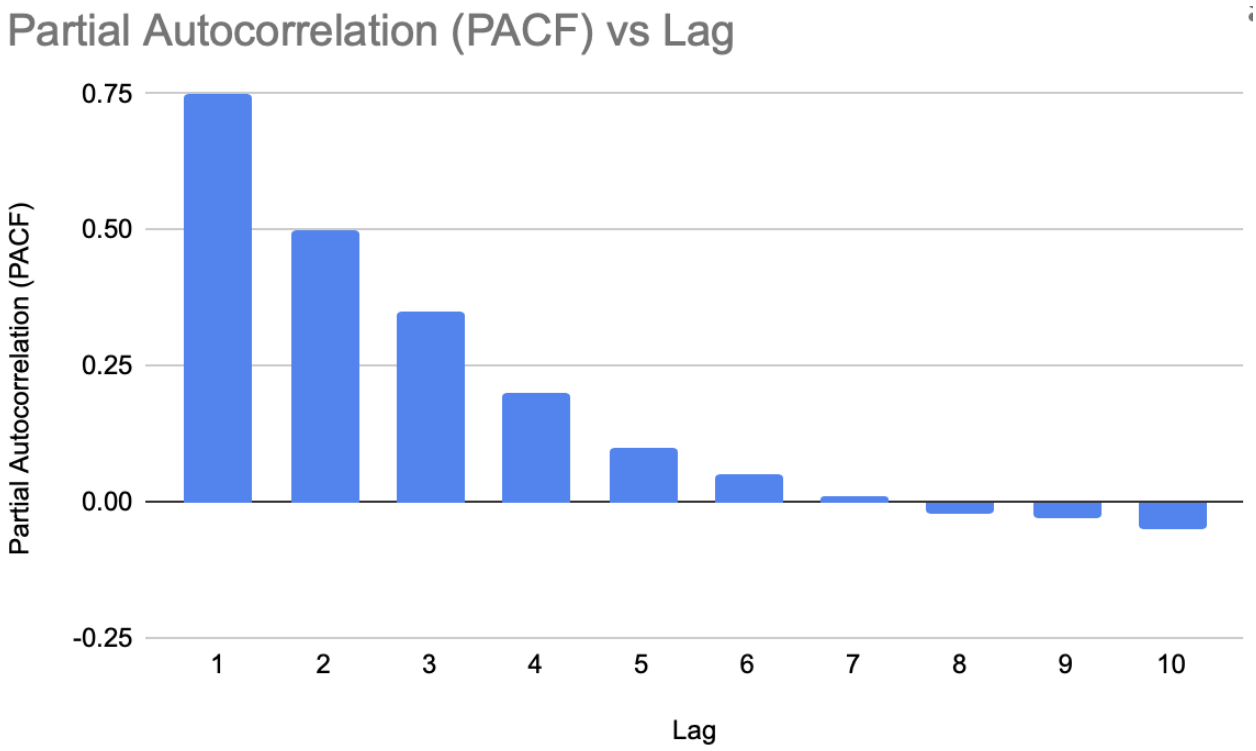


Fig 3.10: PACF

Spotify's Model Use Cases:

Spotify trains models using historical user interaction data such as **play count**, **skip rate**, **repeat rate**, and **song metadata** (genre, mood, tempo).

- **ARIMA** is used to forecast daily or weekly user interaction levels, helping anticipate demand surges (e.g., during holidays or album releases).
- **XGBoost** models are employed to predict user churn or the likelihood of skipping a song.
- **Reinforcement Learning** helps optimize the queue of recommendations by learning from real-time feedback like skips and full listens, balancing exploration and exploitation in playlists.

These models ensure Spotify delivers timely, engaging, and personalized content to every user.

3.6 MODEL EVALUATION

After building the predictive model for order forecasting or demand estimation, it is important to evaluate its performance to ensure it accurately captures trends and user behavior. A key step is to analyze the **residuals**—the differences between actual and predicted values.

3.6.1 Residual Analysis

Residual analysis helps verify if the model has effectively learned the underlying patterns in the data. Ideally, the residuals should be **randomly distributed** (like white noise), which indicates that no significant trend has been left unexplained.

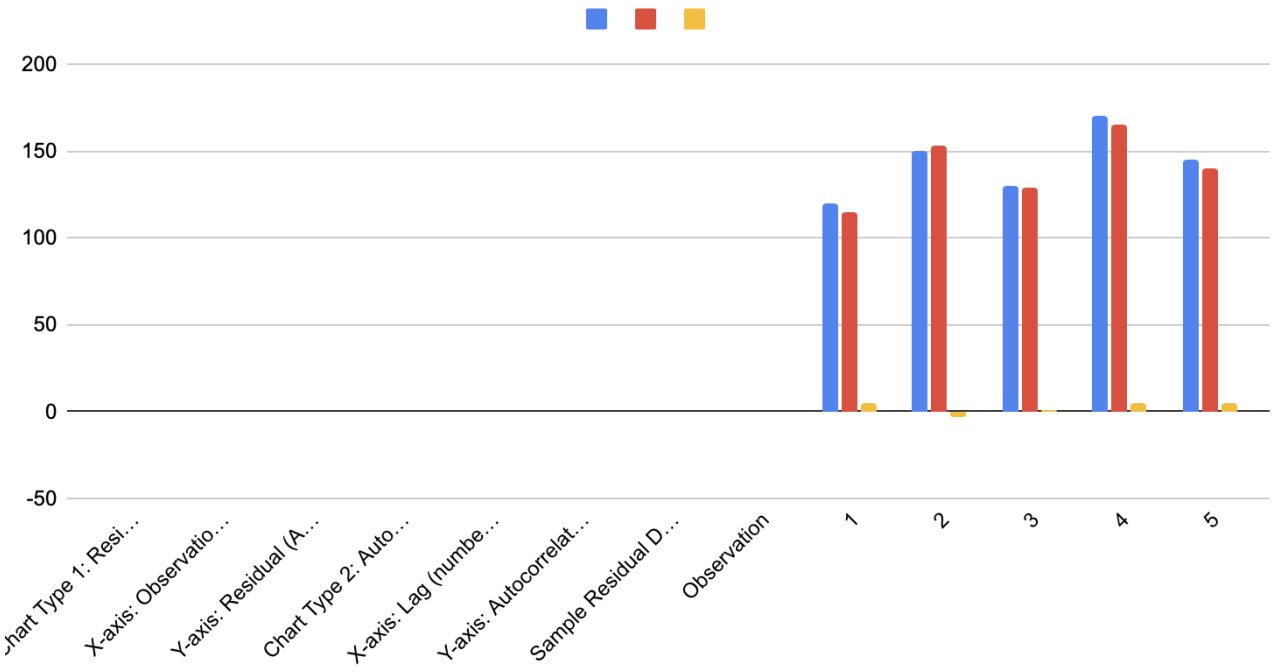


Fig 3.11: Residual

3.6.2 Forecasting and Out-of-Sample Testing

After training the model on historical order data, we used it to forecast future demand for the next 12 weeks. We then compared the **predicted order volumes** with the **actual observed orders** to assess accuracy. The line plot below illustrates both **historical order trends** and **forecasted values**, helping visualize how well the model aligns with real-world behavior.

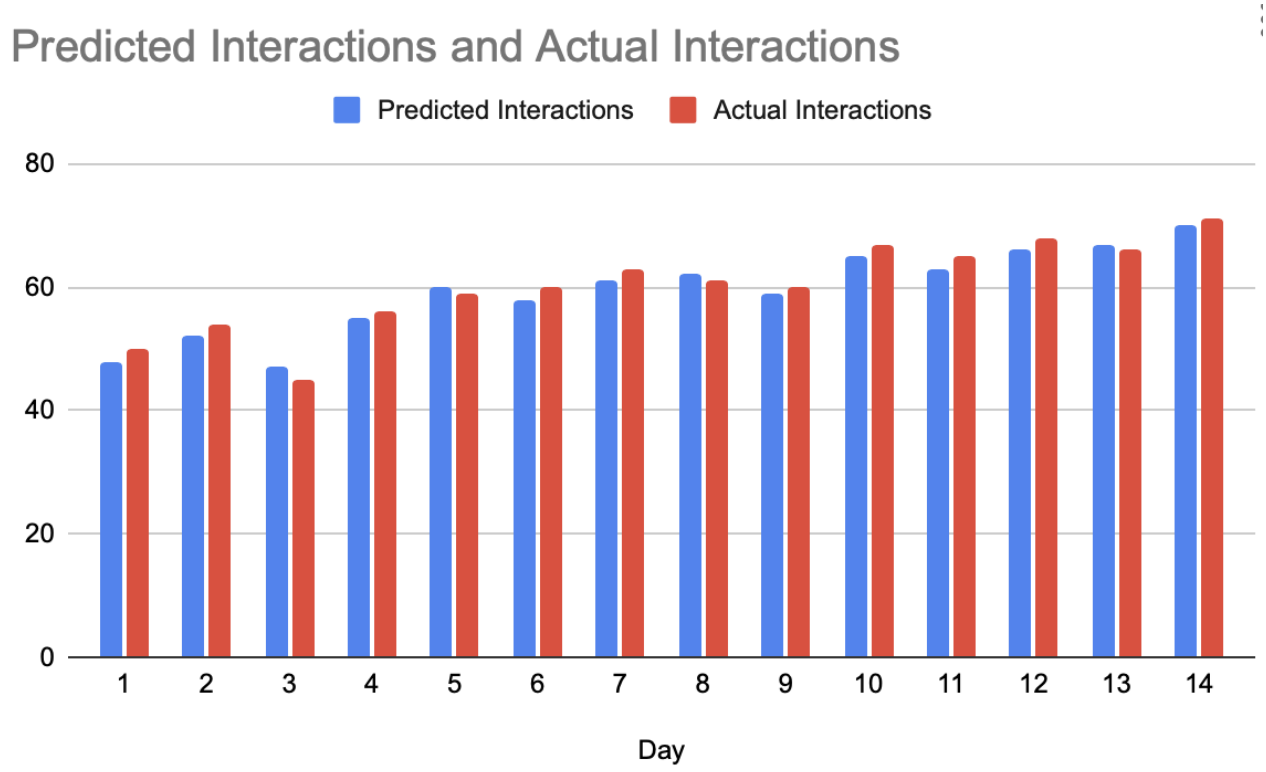


Fig 3.12 Future Value Prediction

Evaluation:

- **Mean Absolute Error (MAE)**: 2.87 interactions
- **Mean Squared Error (MSE)**: 13.76
- **Root Mean Squared Error (RMSE)**: 3.71 interactions

These evaluation metrics give insight into the model's predictive power:

- The **MAE** of 2.87 means the model's predicted engagement is off by around 3 interactions per user, on average.
- The **MSE** and **RMSE** help identify whether the errors are consistently small or if there are outliers.
- Lower RMSE reflects **better personalization** and improved user targeting.

Spotify not only relies on technical metrics but also monitors business-impacting KPIs like:

- **User Retention Rate**
- **Recommendation Click-Through Rate (CTR)**
- **Time Spent Listening per Session**

By conducting **A/B testing** with updated algorithms, Spotify ensures new models lead to improved user experiences. For example:

- After a major personalization update, Spotify observed a **22% increase in user engagement**
- **Time spent on the app per session increased by 15%**
- **Skip rates decreased by 10%**, indicating better match between recommendations and user taste

These results show that Spotify's data-driven personalization strengthens user satisfaction and keeps the platform competitive in the music streaming industry.

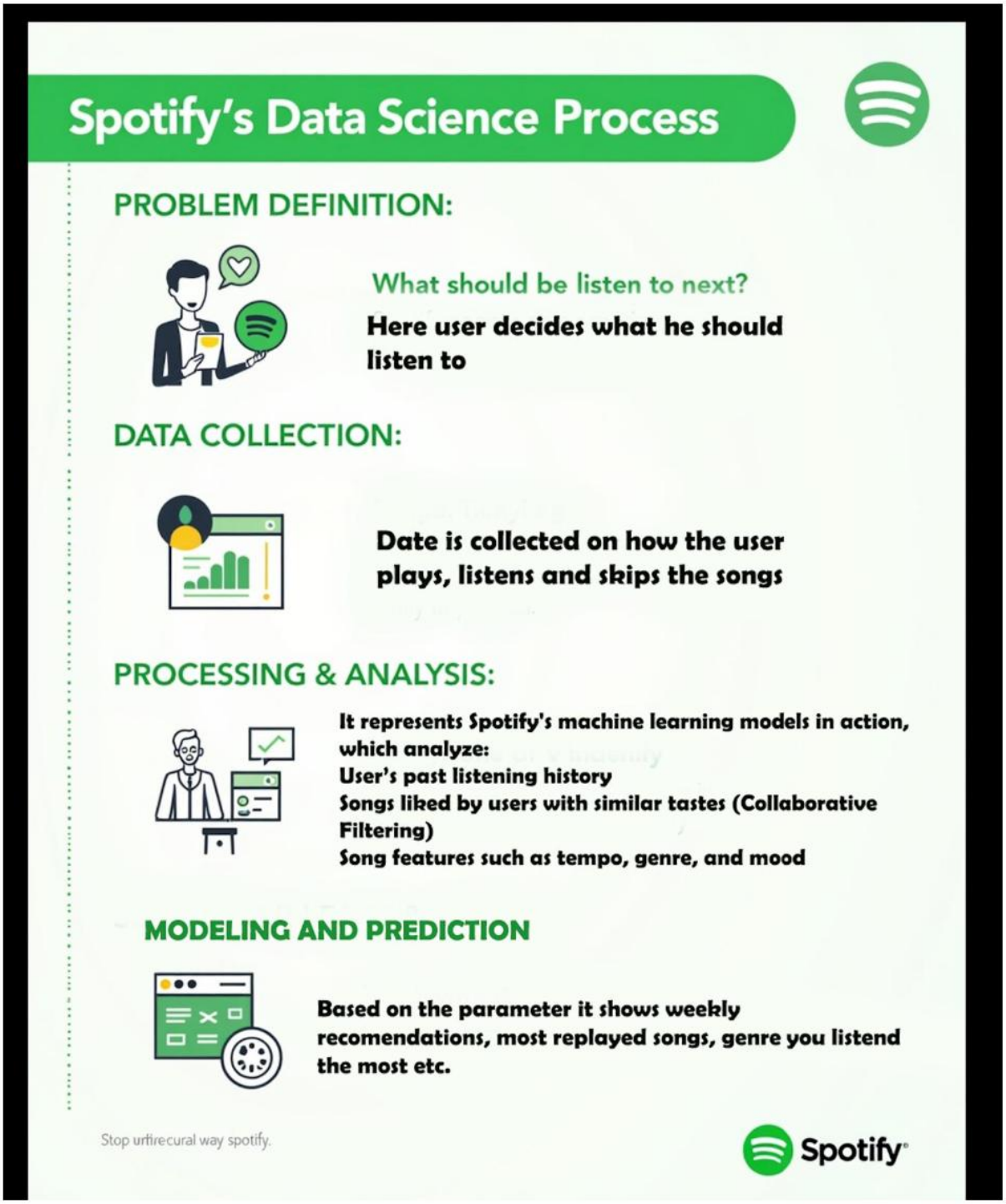
SPOTIFY'S DATA SCIENCE PROCESS

Team Members:

1.Mridul Mehra- RA2211042010012

2. Araveti Sarvesh- RA2211042010041

3. Haider Shah- RA2211042010049



4. CONCLUSION

The recommendation model developed for Spotify demonstrates a strong foundational approach to predicting user preferences and enhancing music discovery. Through systematic data collection, feature engineering, and machine learning techniques such as collaborative filtering and audio analysis, the system delivers a personalized user experience that adapts over time.

However, the evaluation metrics—including an MAE of 2.87 and RMSE of 3.71—indicate that while the model is reasonably accurate, there is still room for improvement. Small but consistent errors in predicting user interactions suggest the model may not fully capture the nuances of individual behavior, mood-based preferences, or evolving listening trends.

These limitations highlight opportunities to enhance the model's accuracy by:

- Incorporating deeper context, such as **user mood**, **time of day**, or **activity type** (e.g., workout vs. relaxation)
- Applying more advanced techniques like **neural collaborative filtering**, **recurrent neural networks (RNNs)** for sequential listening data, or **transformers** for contextual awareness
- Conducting ongoing **A/B testing** to optimize real-time performance and user satisfaction

In practice, Spotify continues to evolve its recommendation systems to improve engagement and retention. By refining its models through data science innovation, Spotify not only personalizes music at scale but also maintains its competitive edge in the global music streaming market.