Analysing User Feedback for Enhanced Music Recommendations

A Project Report

Submitted in partial fulfilment of the requirements for the award of degree of
Master of Computer Applications

LOVELY PROFESSIONAL UNIVERSITY PHAGWARA, PUNJAB

[Aug 2024 to Dec 2024]

Group Number: DC004

Submitted By

Venu Kurugonda(12212481)

Gunda Pavan Sai(12208402)

Pilla Esther Satya(12204465)

Abhinav Raj(12204340)

Kumar Harsh(12204280)

Supervised By

Mr. Ravinder Singh,

Assistant Professor,

School of Computer Application,

Lovely Professional University

LOVELY PROFESSIONAL UNIVERSITY,

PUNJAB

[Aug'24-Dec'24]

Declaration by student

To whom so ever it may Concern

We, **Esther Satya Pilla(12204465)**, **Kurugonda Venu (12212481)**, **Abhinav Raj (12204340)**, **Gunda Pavan Sai (12208402)**, **Kumar Harsh (12204284)** hereby declare that the work done by us on "**User Feedback Analysis for Personalized Recommendations: Improving Spotify's Algorithm through Insights from User Behaviour**" under the supervision of **Ravinder Singh, Asst. Professor**, Lovely professional University, Phagwara, Punjab, is a record of original work for the partial fulfillment of the requirements for the award of the degree, Masters of Computer Applications.

Team: **Signature:**

Esther Satya Pilla (12204465)

Abhinav Raj (12204340)

Gunda Pavan Sai (12208402)

Kurugonda Venu (12212481)

Kumar Harsh (12204284)

Date:

CERTIFICATE

This is to certify that Esther Satya Pilla, Kurugonda Venu, Gunda Pavan Sai, Abhinav Raj, Kumar Harsh has completed doing Capstone Project “User Feedback Analysis for Personalized Recommendations: Improving Spotify's Algorithm through Insights from User Behaviour” under my guidance and supervision. To my knowledge, neither now nor its first discovery. No part of this report has been submitted for any other course or certification. This report is required for the submission and completion of part of the requirements for the award of the degree of Master of Science in computer science.

Signature:

Date:

Mr. Ravinder Singh (28933),

Assistant Professor,

School of Computer Applications,

Lovely Professional University.

Acknowledgement

We thank Ravinder Singh Sir, School of Computer Applications, Lovely Professional University, Phagwara (Punjab). His tireless support, guidance and inspiration were essential to the success of the project. We are particularly grateful to him for his suggestions at key stages of development, his support during difficult times and his patience in explaining difficult topics. His commitment to our education and his belief in our abilities were invaluable throughout the project. Their support, encouragement and confidence in us formed the foundation of this project. We would like to thank them for the excellent results they achieved.

Esther Satya Pilla (12204465)

Gunda Pavan Sai (12208402)

Abhinav Raj (12204340)

Kurugonda Venu (12212481)

Kumar Harsh (12204284)

Table Of Contents

Chapters	Title	Page.no
	Abstarct	1-1
	Introduction	2-2
Chapter 1	Problem Definition and Objectives	3-7
Chapter 2	Literature Review	8-11
Chapter 3	Methodology	12-20
Chapter 4	System Examination	20-23
Chapter 5	System Scheme	23-27
Chapter 6	Implementation	27-36
Chapter 7	Results and Discussion	36-37
Chapter 8	Visualizations	38-38
Chapter 9	Conclusion	39-39
Chapter 10	References	40-40
Chapter 11	Submission of Research Paper	41-41
Chapter 12	Plagiarism report	42-42

Abstract

This Project focuses on identifying hidden patterns and valuable insights from Spotify user feedback to better understand user interaction with the platform. By examining a robust dataset that includes user reviews, subscription behaviour's, and interaction data, the study pinpoints areas for improvement to elevate user experience and foster innovation.

The analysis considers factors such as device preferences, subscription choices, and content consumption to uncover the most popular plans and trending music genres. It further explores the impact of demographic variables like age and gender on listening habits, offering opportunities for user segmentation. These insights guide Spotify developers in refining the platform to be more engaging and tailored to user needs. Ultimately, the findings contribute to product development, enhanced customer support, and informed feature updates, ensuring a more enjoyable and personalized experience for Spotify users.

Keywords: *Spotify, Podcast, Machine learning, Analyze, Predictive analysis, Linear Regression and Decision Tree, Deep Learning, Music, User feedback, subscriptions.*

Introduction

Music streaming services have revolutionized how people access and enjoy music, with Spotify emerging as a dominant platform in this industry. As of 2024, Spotify boasts over 550 million active users per month, including 226 million paying subscribers worldwide. With an extensive collection of more than 100 million tracks and five million podcasts, Spotify has positioned itself as a leader in digital content consumption. This competitive landscape demands continuous improvement in services to ensure user retention and satisfaction.

Spotify collects vast amounts of user data, encompassing listening habits, device preferences, subscription choices, and direct user feedback through ratings and reviews. Analyzing this data allows the platform to identify patterns that drive user engagement, satisfaction, and retention. For instance, insights into popular **subscription plans, trending music genres, and demographic preferences** provide actionable intelligence for enhancing platform functionality and user experience.

This study delves into user behavior by examining variables such as subscription trends, listening devices, and demographic factors like age and gender. The objective is to explore how these variables influence user engagement, satisfaction, and retention. For example, younger audiences may lean toward contemporary music genres, while older users may prefer nostalgic or regional tracks. By understanding these behaviors, Spotify can implement effective personalization strategies that cater to diverse user preferences.

In today's digital era, data is a pivotal resource, and its effective utilization defines success in the competitive market of music streaming. The current research highlights the importance of analyzing user feedback to identify key areas for improvement. With techniques like predictive modeling and machine learning, this study offers insights into enhancing Spotify's algorithms and providing more tailored recommendations. Ultimately, the findings aim to empower Spotify to foster innovation, increase user loyalty, and maintain its position as a leader in the global music streaming sector.

Finally, this study shows that leveraging advanced data analysis techniques, such as machine learning and predictive modeling, can significantly enhance Spotify's recommendation system. By uncovering hidden patterns in user feedback and identifying key factors influencing engagement and retention, Spotify can refine its algorithms to deliver a more personalized and satisfying user experience.

Chapter 1:Problem Definition and Objectives

Music streaming platforms like Spotify rely heavily on user satisfaction to maintain and grow their subscriber base in a competitive market. Despite the extensive use of recommendation algorithms, challenges remain in fully understanding user preferences and addressing gaps in engagement, satisfaction, and retention.

Key issues include:

- Difficulty in catering to diverse demographic groups with varying music preferences.
- Limited insights into the impact of device usage and subscription behavior on user engagement.
- Inefficiencies in identifying potential areas of dissatisfaction from user feedback.

These challenges highlight the need for a more robust and insightful analysis of user feedback, behaviors, and demographics to tailor services effectively and foster innovation (Ahuja, n.d.).

Objectives

The primary objectives of this study are:

1. **Analyze User Feedback**

   Extract hidden patterns from Spotify's user feedback to identify areas needing improvement and innovation.

2. **Understand Demographic Preferences**

   Investigate how factors such as age, gender, and regional diversity influence listening habits and preferences.

3. **Examine Engagement and Retention**

   Explore the relationship between subscription choices, device usage, and user engagement to uncover strategies for improving satisfaction and loyalty.

4. **Enhance Recommendation Systems**

   Utilize machine learning techniques to refine Spotify’s recommendation algorithm, ensuring better personalization and a seamless user experience.

5. **Support Strategic Decision-Making**

Provide actionable insights to aid Spotify in making informed decisions on subscription strategies, and user segmentation.

1.1 Motivation

In the rapidly evolving landscape of digital music streaming, platforms like Spotify have transformed how people access and enjoy music. With over 550 million monthly active users and a highly competitive market, it is crucial for Spotify to continuously innovate and enhance its user experience. The growing diversity in user demographics and preferences underscores the need for more personalized and adaptive solutions.

The motivation for this study stems from the following key factors:

- **Rising User Expectations**

Users increasingly demand tailored experiences that cater to their individual preferences. Failure to meet these expectations can result in dissatisfaction and churn, making it essential to analyze user feedback comprehensively (Javatpoint, n.d.).

- **Technological Advancements**

Advances in data analysis, machine learning, and artificial intelligence provide an unprecedented opportunity to uncover deeper insights into user behavior, paving the way for more effective recommendation systems and platform enhancements.

- **Market Competition**

With multiple music streaming platforms vying for user attention, Spotify's ability to stay ahead depends on understanding and addressing user needs better than competitors.

- **Personal Passion for Innovation**

This study is driven by a passion for leveraging data science to improve user-centric platforms, enhancing both the technical and experiential aspects of Spotify's offerings.

1.2 Purpose

The purpose of this study is to explore and analyze user feedback and behavioral data from Spotify to enhance its recommendation system and overall platform experience. By identifying patterns and trends in user interactions, this research aims to address critical aspects of engagement, satisfaction, and retention in a competitive music streaming industry.

Specifically, the Project seeks to:

1. **Improve User Experience**

   Develop actionable insights that enable Spotify to offer a more personalized and intuitive platform tailored to individual user preferences.

2. **Optimize Recommendation Systems**

   Leverage advanced machine learning models to refine Spotify's recommendation algorithms, ensuring accurate and context-aware music and podcast suggestions.

3. **Enhance User Retention Strategies**

   Identify factors influencing user loyalty and satisfaction, such as subscription preferences and device usage, to formulate strategies that reduce churn and increase premium subscriptions.

4. **Empower Strategic Decisions**

   Provide data-driven insights that inform Spotify's product development, marketing, and customer support strategies.

5. **Support Innovation in Personalization**

   Offer a deeper understanding of demographic impacts on music preferences, helping Spotify implement better segmentation and personalization approaches.

By fulfilling these purposes, the study aims to reinforce Spotify's position as a market leader while delivering a richer, more engaging experience for its users.

1.3 Scope

This study focuses on analyzing user feedback and behavioral data from Spotify to uncover insights that can enhance the platform's recommendation system and user experience. The scope of the research is defined as follows:

1. **Data Analysis**

   The study examines user interaction data, subscription behaviors, and feedback collected from Spotify users. It involves identifying key patterns in listening habits, device preferences, and demographic factors such as age and gender.

2. **Machine Learning Implementation**

   The research incorporates predictive modeling and machine learning techniques, such as Random Forest, Decision Trees, and Gradient Boosting, to analyze user preferences and predict subscription trends (Kaggle).

3. **User Segmentation**

   The study explores how demographic variables influence music consumption, enabling better segmentation of users based on their preferences and behaviors.

4. **Focus on Key Metrics**

   The research emphasizes critical metrics like user engagement, satisfaction, and retention, providing actionable insights for improving these areas.

5. **Recommendation System Enhancement**

   By analyzing existing feedback, the study aims to refine Spotify's recommendation algorithms, ensuring more personalized and relevant suggestions for users.

6. **Demographic and Geographic Diversity**

   The study considers the impact of regional and cultural diversity on user preferences, addressing the unique needs of different user groups across various geographic locations.

While the research primarily focuses on Spotify, its methodologies and insights can be applied to similar streaming platforms, demonstrating broader applicability in the music streaming domain.

1.4 Limitations

While this study aims to provide valuable insights into user behavior and improve Spotify's recommendation system, it is important to acknowledge several limitations that may impact the scope and applicability of the findings:

1. **Data Availability and Quality**

   The study relies on publicly available datasets and user feedback from Spotify. There may be limitations in the comprehensiveness or quality of the data, such as missing information or biased user reviews, which could affect the analysis and results.

2. **Limited Geographic Coverage**

   The research may be constrained by the geographic scope of the dataset, which may not fully represent the preferences and behaviors of all Spotify users worldwide. Demographic trends could vary significantly across different regions, and the study may not capture these variations comprehensively.

3. **Focus on Specific Data Points**

   The analysis focuses on a limited set of variables such as subscription behaviors, device preferences, and demographic factors. While these are important, other factors influencing user engagement, such as social interactions, external recommendations, or platform updates, may not be fully addressed (Madyatmadja, n.d.).

4. **Model Generalization**

   Machine learning models used in the study, such as Random Forest, Decision Trees, and Gradient Boosting, may have limitations in their ability to generalize across different subsets of data. Model accuracy can be affected by factors like data distribution, feature selection, and overfitting, leading to results that may not fully capture the complexity of user behaviors.

5. **External Factors Not Accounted For**

   The study focuses primarily on user feedback and behavioral data within Spotify. However, external factors such as changes in the music industry, new competition, or broader societal trends may influence user behavior but are not considered within the scope of this research.

6. **Potential Bias in User Feedback**

   The feedback collected from users may be skewed based on the nature of the respondents, such as over-representation from certain demographic groups or users with extreme opinions (positive or negative). This may impact the overall accuracy and reliability of the findings.

These limitations suggest that while the findings of this study are valuable, they should be interpreted with consideration of the context and the constraints under which the research was conducted.

Chapter 2:Literature Review

The goal of analyzing user feedback for music recommendation systems is to improve user satisfaction and engagement by understanding their preferences and behavior. Despite advancements in music streaming platforms, challenges remain in delivering personalized recommendations that meet diverse user needs. By leveraging user feedback, machine learning can optimize recommendation algorithms, enhance user experiences, and better predict music preferences.

2.1

Analyzing user feedback provides significant benefits for improving music recommendations, such as identifying listening patterns, understanding user preferences, and personalizing playlists. Machine learning models can process explicit feedback (like ratings or reviews) and implicit feedback (like skips or playtime) to create tailored recommendations. These systems improve user retention by delivering relevant music suggestions, increasing satisfaction through personalization, and identifying emerging trends in user behavior. Overall, analyzing feedback enhances recommendation accuracy, boosts user engagement, and contributes to a more enjoyable listening experience (Ahuja, n.d.).

2.2 Challenges in User Feedback Analysis for Music Recommendations

The primary challenge in analyzing user feedback lies in dealing with noisy, sparse, and biased data, such as incomplete reviews or skewed preferences. Additionally, understanding cultural and contextual factors influencing music choices is complex. Real-time processing of large-scale feedback data requires robust computational resources and scalable models. Ensuring ethical and unbiased recommendations while maintaining user privacy adds another layer of difficulty. Addressing these challenges requires sophisticated modeling techniques and robust data integration strategies.

2.3 Evolution of Recommendation Algorithms

Spotify has evolved from using basic content-based filtering to more advanced collaborative and playlist-based recommendation algorithms. According to Deldjoo et al. (2024), collaborative filtering, which analyzes user preferences based on similarities between users and their interaction with content, is a widely adopted approach. However, more recently, Spotify has incorporated a playlist-centric recommendation model, which looks at how users curate playlists and how songs co-occur within those playlists. This shift enables Spotify to identify

contextual relationships between songs, improving the quality and accuracy of recommendations (Karthik, n.d.).

The effectiveness of these algorithms is further enhanced by the application of deep learning techniques, which can process large volumes of user data to extract complex patterns. A study by Kaur et al. (2023) demonstrated that deep learning models, when applied to user feedback and listening history, could predict user preferences with greater accuracy, leading to more personalized recommendations. This is particularly useful in predicting music genres, moods, or even specific artists that a user may enjoy based on their historical data and feedback.

2.4 Machine Learning and Predictive Analysis in User Feedback

Machine learning and predictive analytics play a crucial role in understanding user behavior and enhancing recommendation systems. Studies such as those by Reddy and Vaghela (2024) emphasize the application of machine learning models, including decision trees, random forests, and gradient boosting, to analyze large datasets and predict user preferences. These models can identify hidden patterns within user feedback, enabling more accurate content recommendations. For instance, using historical feedback data, machine learning models can forecast which users are likely to upgrade to premium plans based on their engagement levels, subscription patterns, and demographic profiles.

Additionally, predictive modeling helps in forecasting user churn, a common challenge for subscription-based platforms. As highlighted by Sood et al. (2023), churn prediction models can identify at-risk users based on their engagement trends, such as decreased listening frequency or negative feedback. This allows Spotify to take proactive measures, such as personalized offers or tailored recommendations, to retain users and reduce churn (Javatpoint, n.d.).

2.5 Best Practices for User Feedback Analysis in Music Recommendations

1. **Multimodal Data Integration:** Combine explicit and implicit feedback, including playtime, skips, ratings, and reviews, for comprehensive user preference modeling.
2. **Collaborative Filtering and Hybrid Models:** Use collaborative filtering, content-based filtering, and hybrid methods to enhance recommendation accuracy.
3. **Context-Aware Analysis:** Incorporate contextual data like time, location, and mood to personalize recommendations dynamically.

4. **Dimensionality Reduction Techniques**: Simplify feature sets while preserving critical information to improve model performance and interpretation.
5. **Federated Learning:** Implement federated learning to build scalable models while ensuring user data privacy.
6. **User-Centric Explanations:** Use interpretable models to explain recommendations, **increasing trust and user satisfaction.**
7. **Cross-Platform Integration:** Collaborate with other platforms (e.g., social media) to access diverse feedback sources for enhanced recommendations.
8. **Continuous Feedback Loop:** Regularly update models with new user feedback to ensure recommendations stay relevant and accurate (Javatpoint, n.d.).

2.6 Existing Systems for Music Feedback Analysis

1. **Traditional Recommender Systems:** Use collaborative or content-based filtering to suggest music based on past behavior, though they may lack personalization for diverse preferences.
2. **Deep Learning Models:** Neural networks and deep learning models analyze vast amounts of feedback data to uncover complex listening patterns.
3. **Real-Time Feedback Platforms:** Platforms like Spotify or YouTube leverage real-time user interactions (e.g., skips, likes) to refine recommendations instantly.
4. **Sentiment Analysis Systems:** Text-based feedback systems analyze reviews or comments to gauge user sentiment towards songs or playlists.
5. **Context-Aware Recommendation Engines:** Models that consider contextual data such as time of day or activity to suggest relevant music.
6. **Social Listening Platforms:** Analyze user discussions on social media to identify popular trends and preferences.
7. **Feedback-Driven Machine Learning Models:** Use user behavior metrics, such as repeat listens and playlist additions, to train personalized recommendation algorithms.

By combining advanced machine learning techniques with continuous user feedback, music recommendation systems can evolve into highly personalized and context-aware platforms, offering an unparalleled listening experience (Karthik, n.d.).

2.8 Proposed Systems

To enhance music recommendations through user feedback analysis, the proposed systems integrate advanced machine learning techniques and user-centric approaches. These systems aim to address the limitations of existing methods by incorporating real-time feedback, improving personalization, and ensuring ethical and unbiased recommendations.

1. **Real-Time Feedback Integration:** Continuously analyze user interactions, such as skips, likes, and playtime, to refine recommendations dynamically.
2. **Hybrid Recommender Systems:** Combine collaborative filtering, content-based filtering, and deep learning to deliver more accurate and diverse recommendations.
3. **Context-Aware Models**: Leverage contextual data like mood, time of day, and location to suggest music tailored to user situations.
4. **Sentiment Analysis for Explicit Feedback:** Use natural language processing to analyze user reviews and comments, capturing sentiment to enhance song recommendations.
5. **Privacy-Preserving Techniques**: Implement federated learning to process user data securely while maintaining personalization.
6. **Behavioral Pattern Recognition:** Use advanced machine learning models to detect listening habits, trends, and emerging preferences.
7. **Cross-Platform Feedback Aggregation:** Integrate data from multiple sources, such as social media or third-party apps, to improve the breadth and accuracy of **recommendations.**
8. **Dynamic Playlist Generation**: Create playlists that adapt to user feedback in real time, ensuring relevance and satisfaction.
9. **Explainable AI Systems:** Develop interpretable models that provide users with understandable insights into why certain songs are recommended.

10. **Continuous Learning Framework:** Regularly update algorithms with new feedback and trends to ensure recommendations stay current and accurate (Ahuja, n.d.).

These proposed systems aim to create a seamless, personalized, and adaptive music recommendation experience, fostering greater user engagement and satisfaction.

Chapter 3: Methodology

This chapter outlines the methodology used to carry out the research for improving Spotify's recommendation system through user feedback analysis. The steps include data collection, preprocessing, model creation, model evaluation, and visualization. Each of these stages is crucial in obtaining meaningful insights from the data and refining the system for better user personalization.

3. 1 Data Collection

Data collection is the first and essential step in this research. The dataset used in this study is sourced from publicly available repositories, primarily Kaggle, which hosts a rich collection of user behavior data related to Spotify. This data includes user interactions, feedback, subscription preferences, device usage, and demographic information.

The key elements of the dataset include:

- **User Reviews**: Textual feedback from users about their experiences with Spotify, such as ratings, comments, and complaints.
- **Subscription Data**: Information on user subscriptions, such as whether the user has a free, premium, or family plan.
- **Listening Habits**: Data on the songs, genres, and artists that users listen to most frequently.
- **Demographics**: Information such as age, gender, and geographical location of the users.
- **Device Preferences**: Insights into which devices (e.g., smartphone, tablet, laptop) are most commonly used for accessing Spotify.

The data was collected with a focus on a representative sample of Spotify users to ensure diversity in terms of age, gender, region, and subscription type (Kaggle).

3.2 Data Preprocessing

Data preprocessing is a crucial step to ensure that the raw data is clean, consistent, and suitable for analysis. The preprocessing phase involves several steps:

1. **Handling Missing Data**

   Missing values are identified and handled using imputation techniques, where feasible. For instance, if a demographic feature (e.g., age or gender) is missing, it might be filled with the mode or median value, depending on the distribution. In cases where a feature is missing for too many users, it may be excluded from the analysis.

2. **Data Cleaning**

   Outliers and inconsistent data points are detected using visualizations such as box plots. For example, extreme values in subscription duration or listening hours are flagged and treated, either by removing or adjusting them based on domain knowledge.

3. **Encoding Categorical Data**

   Categorical variables such as subscription type (e.g., free, premium) and device preferences (e.g., smartphone, laptop) are encoded into numerical values using techniques such as one-hot encoding or label encoding.

4. **Feature Engineering**

   New features are created based on existing data to capture additional insights. For example, combining age and subscription plan may provide a better indication of user engagement trends. Similarly, a feature might be created based on a user's most preferred genre or artist to give more weight to those preferences in the recommendation model (Karthik, n.d.).

3.3 Model Creation

In this step, machine learning models are built to analyze user behavior and predict preferences for music recommendations. Several models are used to compare their performance and choose the best-performing one for personalization:

1. **Decision Trees**

A Decision Tree algorithm is used to model decision-making based on user features such as age, gender, and subscription type. The tree structure is used to segment users into different groups based on their preferences, helping to predict what kind of music they might enjoy next.

2. **Random Forest**

   Random Forest is an ensemble method that combines multiple decision trees to improve prediction accuracy and prevent overfitting. It is particularly effective in handling large, complex datasets, as it reduces the variance associated with a single decision tree model.

3. **Gradient Boosting**

   Gradient Boosting is used to improve prediction accuracy by combining weak learners (decision trees) and adjusting them sequentially to minimize errors. It is particularly effective in handling imbalanced datasets, where certain user behaviors or preferences might be underrepresented.

4. **Support Vector Machines (SVM)**

   SVM is employed to classify user preferences into different categories, such as music genres or subscription types. The SVM algorithm helps find the hyperplane that maximizes the margin between different user groups, ensuring accurate classification of user behavior.

5. **Linear Regression**

   Linear regression is a fundamental machine learning model used to identify and understand linear relationships between variables. This model assumes a straight-line relationship between independent variables (features) and the dependent variable (target). It is particularly effective for predicting outcomes based on one or more features, such as estimating user engagement levels based on factors like age, time spent on a platform, or number of interactions. While it is not suitable for capturing complex nonlinear patterns, linear regression provides valuable insights into the strength and direction of simple relationships within user data (Javatpoint, n.d.).

Each model is trained using the pre-processed dataset, with training and testing data splits to ensure that the models generalize well to unseen data.

3.4 Technologies Used

This section outlines the various technologies and tools employed in the methodology for analyzing user feedback, training models, and evaluating the performance of the proposed recommendation system. The key technologies used in this study include Python, Jupyter Notebook, and various libraries for data exploration, feature engineering, model training, and visualization, apart from this using HTML, CSS, JavaScript to make a dashboard for visual representation on webpage.

Python

Python is the primary programming language used in this study due to its powerful data analysis, machine learning, and visualization capabilities. Python is widely used in data science and machine learning because of its extensive libraries and frameworks that streamline tasks such as data preprocessing, model creation, and evaluation. The following Python libraries were utilized:

- **Pandas**: Used for data manipulation and analysis, such as reading data, cleaning, and handling missing values.
- **NumPy**: Used for numerical operations, particularly for array handling, statistical computations, and mathematical transformations.
- **Scikit-learn**: This machine learning library provides various algorithms for classification, regression, and clustering. It was used for building and evaluating models like Decision Trees, Random Forests, Gradient Boosting, and more.
- **TensorFlow/Keras**: Used for implementing deep learning models such as feed-forward neural networks for user preference prediction.
- **Matplotlib and Seaborn**: These libraries were used for data visualization, including creating bar charts, histograms, box plots, and heatmaps to analyze the relationships between features and visualize model performance.
- **XGBoost**: A popular machine learning framework for gradient boosting, it was used to improve model accuracy and optimize prediction results for complex user behavior analysis.

Python's flexibility and rich ecosystem of libraries make it an ideal tool for implementing data-driven research in this project (Javatpoint, n.d.).

Data Priors

Data priors refer to the assumptions or beliefs we have about the data before observing it. These assumptions are based on domain knowledge or past research, and they influence the data collection and preprocessing steps. In this study, the following priors were considered:

1. **Demographic Factors**: It is assumed that user preferences are influenced by demographic factors such as age, gender, and geographical location. Younger users may prefer contemporary music genres, while older users may lean towards nostalgic or classic music.
2. **User Behavior**: Users who have been on the platform longer may demonstrate stronger preferences for specific genres, artists, or subscription plans.
3. **Device Usage**: It is assumed that the type of device used (smartphone, tablet, laptop) may impact music consumption patterns, such as the time of listening and the likelihood of subscribing to premium services.
4. **Subscription Types**: It is hypothesized that users with premium subscriptions tend to engage more with the platform, listen to more diverse content, and have higher levels of satisfaction compared to free-tier users.

These data priors guided the creation of features and the analysis of user feedback to uncover insights that align with known trends in the streaming industry.

Feature Engineering

Feature engineering involves creating new features from raw data that can improve model performance and make the analysis more insightful. In this study, the following feature engineering techniques were applied:

1. **User Engagement Features**: Features such as "total listening hours," "average song duration," and "playlist creation frequency" were created to capture how engaged a user is with the platform.
2. **Subscription Behavior**: Features were created to identify trends in subscription behavior, such as "subscription type" (free, premium, family) and "upgrade likelihood" (based on demographic and behavioral patterns).

3. **Content Preferences**: Features like "favorite genres" and "top artists" were derived from user listening history to enhance recommendations based on content preferences.
4. **Time-Based Features**: Features like "time of day" and "seasonality" (e.g., usage patterns during holidays) were incorporated to personalize recommendations based on when the user typically engages with the platform.
5. **Interaction with User Feedback**: Sentiment analysis of user reviews was used to create features reflecting user sentiment, such as "positive feedback score" or "dissatisfaction score."

These features were engineered to help the models understand complex patterns in user preferences and behavior, leading to more accurate predictions and personalized recommendations.

Model Training

Model training involves feeding the preprocessed data into machine learning algorithms to learn patterns and relationships that can predict future outcomes. The following steps were involved in model training:

1. **Splitting the Dataset**: The dataset was divided into training and testing sets. Typically, 70-80% of the data was used for training, and the remaining 20-30% was reserved for testing the model's performance.
2. **Training Models**: Different machine learning models, including Decision Trees, Random Forests, Gradient Boosting, and Deep Learning models, were trained on the dataset. Each model was trained using cross-validation to optimize hyperparameters and prevent overfitting.
3. **Model Fitting**: Once the optimal hyperparameters were selected, the models were trained on the entire training dataset to fit the best possible model to the data (Javatpoint, n.d.).

Model Evaluation and Validation

Once the models were trained, they were evaluated using various metrics to assess their performance. The following evaluation techniques were applied:

1. **Accuracy**: This metric measured the overall correctness of the model in predicting user behavior, especially for classification tasks like subscription prediction.
2. **Precision, Recall, and F1-Score**: These metrics were used to evaluate the performance of classification models, particularly in distinguishing between users likely to upgrade to premium versus those who would remain on the free plan.
3. **Mean Squared Error (MSE) and R-Squared**: For regression tasks, MSE was used to measure the difference between predicted and actual values. R-squared was used to assess how well the model explained the variance in the user data.
4. **Cross-Validation**: K-fold cross-validation was used to assess the model's ability to generalize to new, unseen data and reduce the risk of overfitting. Each fold was used to test the model, providing a more robust evaluation of its performance.
5. **ROC-AUC Curve**: For binary classification problems (e.g., predicting premium subscription likelihood), ROC and AUC scores were used to evaluate the model's ability to distinguish between different classes.

Jupyter Notebook

Jupyter Notebook was the development environment used for the research. It provided an interactive interface for data analysis, model training, and evaluation. Key advantages of using Jupyter include:

1. **Data Exploration and Analysis**: Jupyter allows for real-time interaction with data, enabling step-by-step exploration, data cleaning, and transformation.
2. **Model Prototyping**: Jupyter Notebook provides an easy platform for quickly prototyping machine learning models and evaluating their performance using various metrics.
3. **Visualization Integration**: The integration of Python libraries like Matplotlib, Seaborn, and Plotly made it easy to generate interactive visualizations, aiding in the interpretation of results and the communication of insights.
4. **Collaboration**: Jupyter Notebooks allow easy sharing and collaboration among team members, ensuring that all steps are reproducible and transparent (Kaggle).

Data Exploration

Data exploration involves understanding the structure and patterns within the data before applying machine learning models. In this study, data exploration steps included:

1. **Descriptive Statistics**: Key statistics such as mean, median, and standard deviation were calculated for numeric features to understand their distribution.
2. **Correlation Analysis**: The correlation between various features, such as subscription types, device usage, and listening habits, was explored to identify strong relationships.
3. **Visualization**: Various plots, including histograms, box plots, and scatter plots, were used to visually explore the data and identify potential outliers or patterns.

Prototyping Models

In this phase, multiple models were prototyped using the training dataset. The following machine learning algorithms were prototyped and compared:

1. **Decision Trees**: Simple yet interpretable models used to classify users based on features like demographics and subscription behavior.
2. **Random Forest**: An ensemble method combining multiple decision trees to improve accuracy and reduce variance.
3. **Gradient Boosting**: A boosting algorithm that sequentially corrects errors in the previous models, increasing the accuracy of predictions.
4. **Linear Regression:**
   A simple and interpretable algorithm that models the relationship between independent variables and a dependent variable by fitting a straight line, making it effective for predicting continuous outcomes.
5. **K-Nearest Neighbors (KNN):**
   A non-parametric algorithm that predicts outcomes based on the similarity of data points, identifying the ‘k’ closest neighbors in the feature space to make classifications or predictions.

Data and Visualization

Data visualization is crucial for understanding trends and patterns in the dataset. The following visualization techniques were used:

1. **Bar Charts and Histograms**: To display distributions of categorical and numerical variables, such as subscription types and age groups.
2. **Heatmaps**: To show correlations between features, highlighting the most influential factors affecting user behavior.
3. **Pie Charts:**
   Pie charts visually represent the proportion of categories, making them ideal for showing the distribution of user preferences, such as device usage or subscription plans.

Chapter 4: System Examination

This chapter provides an in-depth examination of the system's requirements, both functional and non-functional, for the proposed recommendation system designed to enhance Spotify's music recommendation algorithm based on user feedback analysis. The chapter outlines the system's capabilities and constraints, focusing on how the system is expected to perform in terms of data collection, prediction generation, and result presentation, as well as its scalability, security, and usability.

4.1 Requirement Analysis

Functional Requirements:

1. Input Data Collection:
   ● The system should provide an intuitive interface for users to input music preferences, feedback, and interaction details, such as genre preferences, favorite artists, skip rates, like/dislike actions, and listening time.
   ● Information verification should ensure that user-provided data is accurate and complete, minimizing noise in the feedback dataset.
2. RecommendationGeneration:
   ● After receiving input, the system should leverage machine learning models to analyze user feedback and generate personalized music recommendations.
   ● Recommendations should be clear, highlighting genres, artists, or playlists that align with user preferences.
3. ResultPresentation:
   ● The system should present recommendations visually and intuitively, using

elements such as charts, playlists, and mood-based categorization.

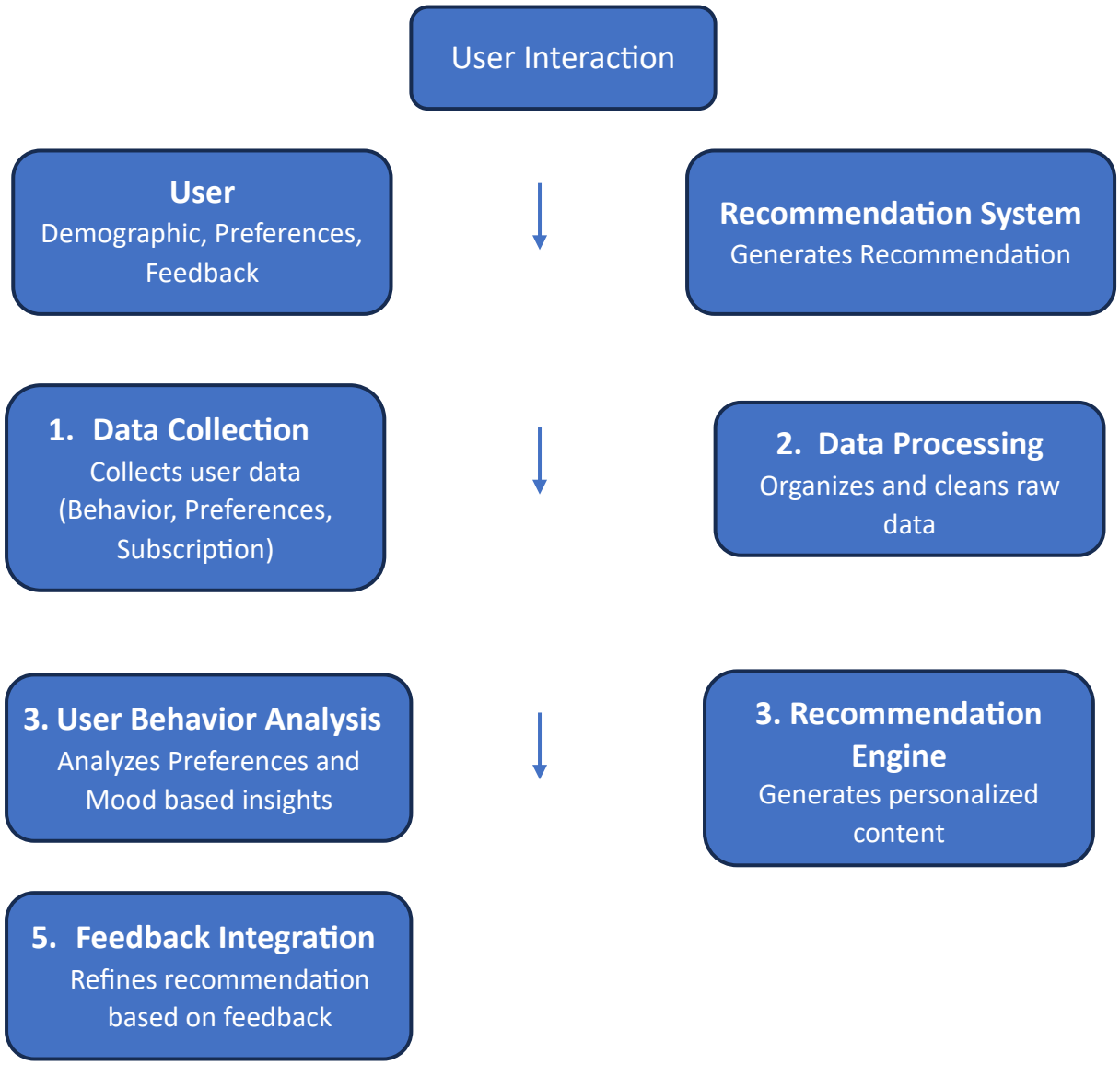
● Results should include personalized playlists, suggestions for exploring new genres, and explanations of why specific recommendations were made (Karthik, n.d.).

Non-Functional Requirements:

1. Performance:
   ● The system must handle multiple user requests efficiently without noticeable delays.
   ● Recommendations should be generated with minimal response time for an optimal user experience.
2. Scalability:
   ● The system should scale seamlessly to accommodate increasing users and larger volumes of feedback data.
   ● Scalability considerations must include computational resources for machine learning models and data storage capacity.
3. Security:
   ● Robust security measures must be implemented to safeguard user data and ensure data privacy.
   ● Encryption should be used to protect user information from unauthorized access or tampering.
4. Reliability:
   ● The system must be reliable, ensuring minimal downtime and consistent functionality.
   ● Mechanisms should monitor system health and alert administrators in case of issues or failures.
5. Usability:
   ● The user interface should be intuitive, user-friendly, and suitable for individuals with varying levels of technical proficiency.
   ● Tutorials or onboarding features should be included to help users navigate the system easily (Karthik, n.d.).

6. Compliance Monitoring:
   - The system should obtain explicit user consent for data processing, ensuring transparency in how data is used.
   - Procedures should be implemented for users to request data access, modification, or deletion in compliance with data protection regulations.

4.1 Data Flow Diagram:



4.2 Hardware Specification:

Processor – Intel

I5 10^{th} Gen or

Latest. RAM 8GB

Hard disk – 30 GB

4.3 Software Specification:

1. Python:
    - Version: Python 3.7
    - Description: Python is the primary programming language for database, model learning, and reverse engineering.

2. Jupyter Notebook:
    - Version: Notebook v6
    - Description: Jupyter Notepad provides a smart place to explore data, present prototypes, and documentation.

3. Html, CSS, JavaScript:
    - Version: HTML5, ES6
    - Description: Dashboard is being created to represent the visual representation of our data analysis (Javatpoint, n.d.).

Chapter 5: System Scheme

This chapter outlines the framework of the proposed system for improving Spotify's recommendation algorithm through user feedback analysis. It describes the flow of data through the system, from acquiring the data to generating predictions and integrating the recommendation model into the overall system architecture. The chapter details the key components of the system, including data acquisition, preprocessing, model creation, evaluation, and the integration of the model into the Spotify platform.

5.1 System Framework

The system framework is designed to process large volumes of user data, apply machine learning algorithms for prediction generation, and integrate the results into a seamless user experience. The framework follows a systematic approach to ensure that the recommendations are accurate, personalized, and scalable.

The system operates in five main stages, as outlined below:

1. Data Acquiring:

- Multiple datasets from Kaggle, including user feedback, music preferences, listening

history, and contextual information (e.g., time of day, mood), were considered.
● Relevant data points, such as genre preferences, skip rates, like/dislike actions, listening duration, and demographic information, were selected based on their relationship to user behavior and music preferences.

2**. Data Preprocessing:**

● Data was cleaned and prepared to ensure compatibility with the system design.
● Preprocessing techniques included:
● Encoding categorical data (e.g., genre, mood) into numerical labels.
● Using libraries like NumPy and Pandas to handle imbalanced datasets and ensure uniform data distribution.
● Addressing missing or incomplete feedback data through imputation or removal.

3. **Model Creation:**

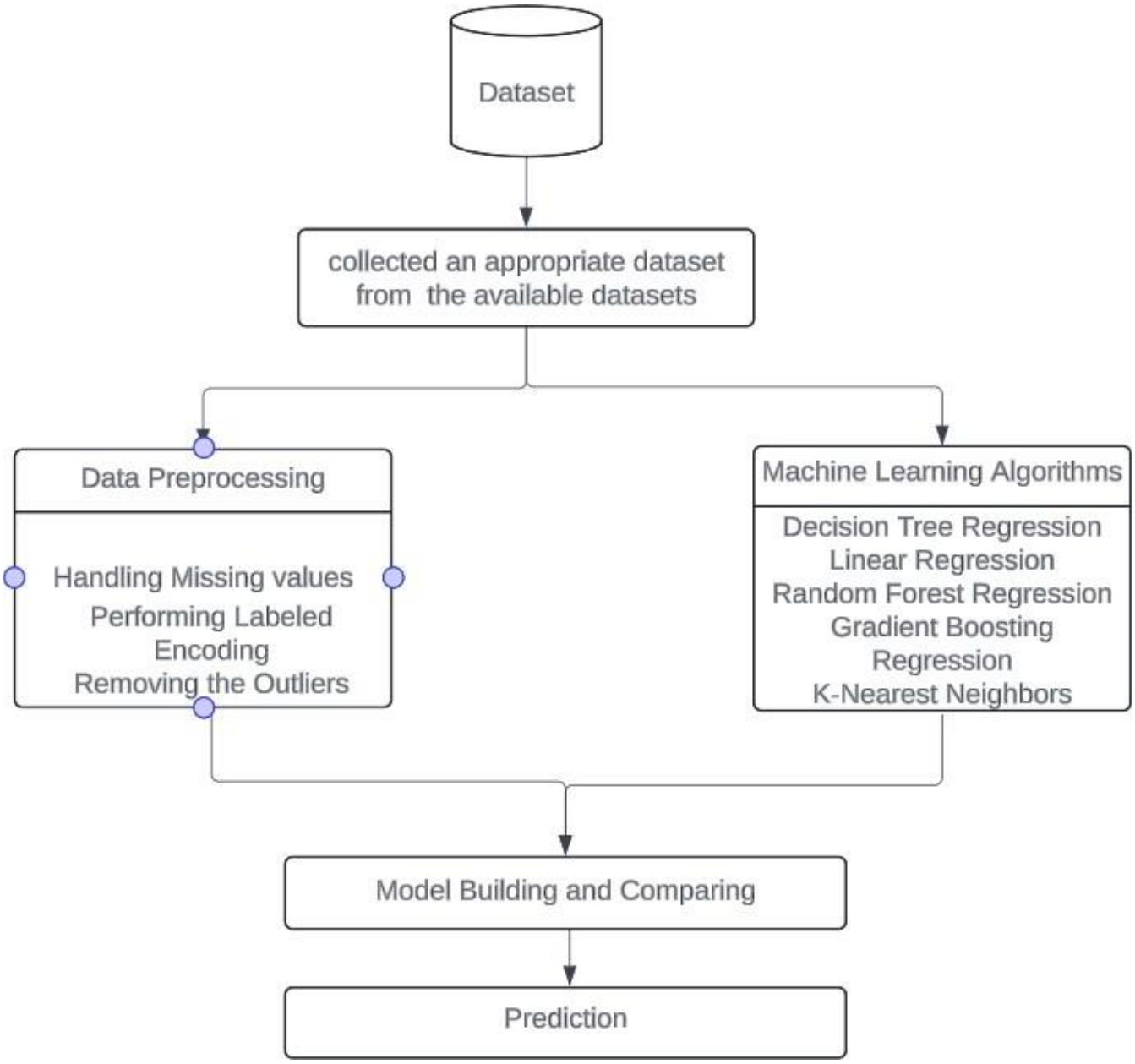
● Machine learning models were developed using the processed data to analyze user feedback and generate recommendations.
● Techniques included Collaborative Filtering, K-Nearest Neighbors (KNN), Random Forest, Gradient Boosting, Neural Networks, and Linear Regression.

4**. Model Evaluation:**

● Six different models were evaluated using performance metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared.
● The model with the best balance of accuracy and scalability was selected for deployment.

5. **Integration and System Process:**

● All components, including data acquisition, preprocessing, model training, and recommendation generation, were integrated into a unified system.
● The system ensures seamless communication between components, enabling real-time recommendation updates.
● A visualization pipeline was designed to present the workflow and results effectively, including charts and graphs that showcase recommendations, trends, and user insights (Kaggle).


Dataset
collected an appropriate dataset from the available datasets
Data Preprocessing
Handling Missing values
Performing Labeled Encoding
Removing the Outliers
Machine Learning Algorithms
Decision Tree Regression
Linear Regression
Random Forest Regression
Gradient Boosting Regression
K-Nearest Neighbors
Model Building and Comparing
Prediction

5.2 Dataset

Dataset includes several columns that capture **user preferences**, **demographics**, and **music listening habits**, all of which are essential for improving Spotify's recommendation system.

Key Columns from the Dataset:

1. **Demographic Information**:
    - **Age**: Represents the user's age group (e.g., 20-35).
    - **Gender**: Gender of the user (all listed as "Female" in your sample).
2. **Spotify Usage Details**:
    - **spotify_usage_period**: Duration the user has been using Spotify (e.g., More than 2 years).
    - **spotify_listening_device**: Device used for listening (e.g., Smart speakers, Computer, Smartphone).
    - **spotify_subscription_plan**: Subscription plan (e.g., Free or Premium).
    - **premium_sub_willingness**: Whether the user is willing to upgrade to a premium subscription (Yes or No) (Kaggle).
3. **Music Preferences**:
    - **preferred_premium_plan**: The preferred premium plan if the user decides to upgrade.
    - **preferred_listening_content**: The preferred type of content (e.g., Podcast, Music).
    - **fav_music_genre**: Favorite genre of music (e.g., Melody, Pop, Rap).
    - **music_time_slot**: Preferred time for listening to music (e.g., Night, Afternoon).
    - **music_influencial_mood**: The mood the music is meant to influence (e.g., Relaxation, Uplifting, Social gatherings).
    - **music_lis_frequency**: Frequency of listening to music (e.g., Leisure time, Office hours) (Kaggle).
4. **Podcast Preferences**:
    - **pod_lis_frequency**: Frequency of podcast listening (e.g., Daily, Weekly).
    - **fav_pod_genre**: Favorite podcast genre (e.g., Comedy, Lifestyle).
    - **preferred_pod_format**: Preferred podcast format (e.g., Storytelling, Interviews).

- **pod_host_preference**: Preference for podcast hosts (e.g., Well-known individuals).
- **preferred_pod_duration**: Desired podcast length (e.g., Shorter, Longer).
- **pod_variety_satisfaction**: Satisfaction with the variety of podcasts (e.g., Ok, Satisfied).

Insights for Model Creation and Evaluation

With this dataset, your project aims to improve the recommendation system by analyzing:

- **User Preferences** (music genre, time of day, podcast format, etc.)
- **Demographics** (age, gender)
- **Subscription Choices** (premium willingness, plan preferences)
- **Device Usage** (which devices are most commonly used)

Using this dataset, you can create models to predict:

- The **likelihood of a user upgrading to a premium plan** based on their current behavior.
- **Personalized music and podcast recommendations** based on user preferences and mood influences (Kaggle).

Chapter 6: Implementation

6.1 Coding:

6.1.1 Data Preprocessing:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
# Load data from CSV file
df = pd.read_csv("D:/datasets/Spotify_data1.csv")
df.head()
```

	Age	Gender	spotify_usage_period	spotify_listening_device	spotify_subscription_plan	premium_sub_willingness	preffered_premium_plan	preferred_listening_content
0	20-35	Female	More than 2 years	Smart speakers or voice assistants	Free (ad-supported)	Yes	Family Plan-Rs 179/month	Podcast
1	Dec-20	Female	More than 2 years	Computer or laptop	Free (ad-supported)	Yes	Individual Plan- Rs 119/ month	Podcast
2	35-60	Female	6 months to 1 year	Smart speakers or voice assistants	Free (ad-supported)	Yes	Student Plan-Rs 59/month	Podcast

6.2 Handling Missing values:

```
# Option 2: Impute missing values (e.g., with mode for categorical columns)
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder
imputer = SimpleImputer(strategy='most_frequent', fill_value='unknown')
df[['preffered_premium_plan', 'fav_pod_genre', 'preffered_pod_format',
    'pod_host_preference', 'preffered_pod_duration','preferred_listening_content']] = imputer.fit_transform(df[
        ['preffered_premium_plan', 'fav_pod_genre', 'preffered_pod_format',
         'pod_host_preference', 'preffered_pod_duration','preferred_listening_content']])


# Check if 'Gender' is already categorical
if not pd.api.types.is_categorical_dtype(df['Gender']):
    categorical_columns = [col for col in df.columns if df[col].dtype == 'object']
else:
    categorical_columns = [col for col in df.columns if df[col].dtype == 'object' and col != 'Gender']


# Encode categorical columns with LabelEncoder
le = LabelEncoder()
for col in categorical_columns:
    df[col] = le.fit_transform(df[col])
```

Outliers Treatment using Capping Approach

```
# Finding the Quantiles
Q1 = sub_table_loc['spotify_listening_device'].quantile(0.25) # First Quartile (25th percentile)
Q2 = sub_table_loc['spotify_listening_device'].quantile(0.50) # Second Quartile (50th percentile, median)
Q3 = sub_table_loc['spotify_listening_device'].quantile(0.75) # Third Quartile (75th percentile)
# IQR: Inter-Quartile Range
IQR = Q3 - Q1
# Lower Limit
LC = Q1 - (1.5 * IQR
```

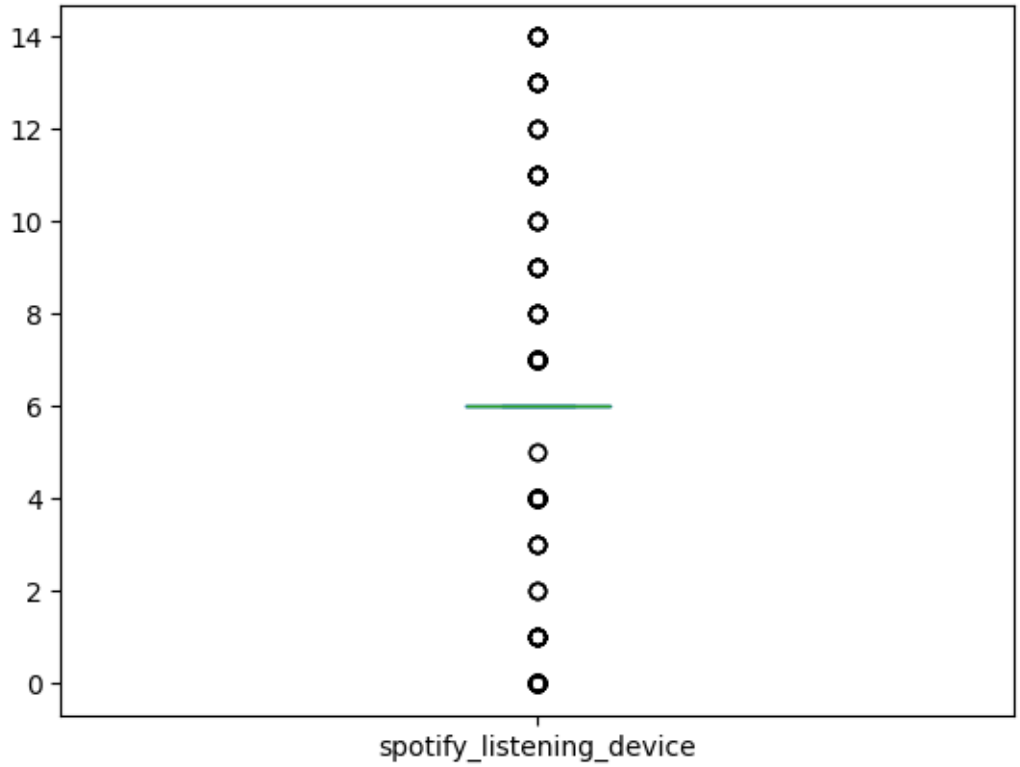
 # Upper Limit

UC = Q3 + (1.5 * IQR)

 # Display the lower and upper limits

print("Lower Limit:", LC)

print("Upper Limit:", UC)



Plot

sns.distplot(sub_table_loc.spotify_listening_device)

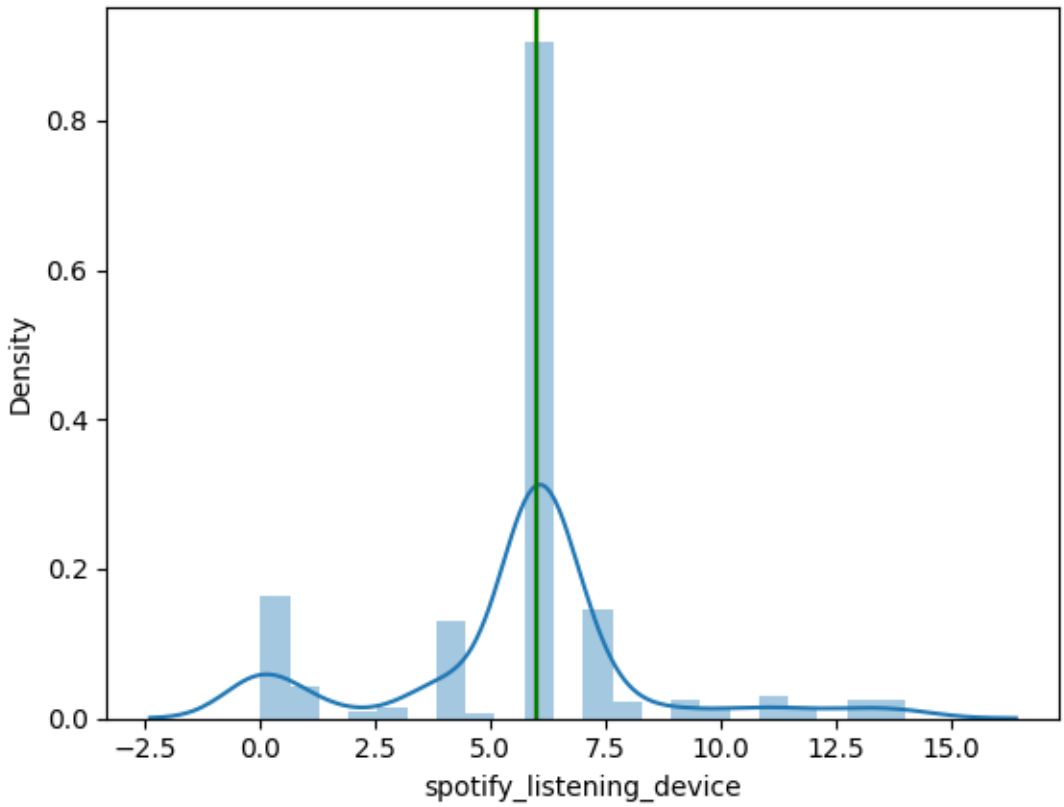
plt.axvline(UC, color='r')

plt.axvline(LC, color ='r')

plt.axvline(Q1, color='g')

plt.axvline(Q3, color='g')

plt.show()



1) Premium Subscription by Gender

Declare independent and target variables for Spotify useage:

 # Assuming the target is a combination of both columns

df3['spotify_useage_by_gender'] = (df3['Gender'] + df3['premium_sub_willingness']) / 2

 # Feature selection - use the correct column names

features = df3[['Gender', 'premium_sub_willingness']]

target = df3['spotify_useage_by_gender'] (Javatpoint, n.d.)

6.3 Linear Regression Model:

import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

```
from sklearn.metrics import mean_squared_error, r2_score
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X,Y, test_size=0.2, random_state=42)
# Check shapes after split
print("X_train shape:", X_train.shape)
print("y_train shape:", y_train.shape)
X_train shape: (415, 2)
y_train shape: (415,)


# Fit the model
model = LinearRegression()
model.fit(X_train, y_train)
# Make predictions
y_pred = model.predict(X_test)
# Calculate metrics
mse_rf = mean_squared_error(y_test, y_pred)
r2_rf = r2_score(y_test, y_pred)
# Evaluate the Linear Regression model
#from sklearn.metrics import mean_absolute_error as mas
y_pred = model.predict(X_test)
print('Linear Regression MAE:', mean_squared_error(y_test, y_pred))
print('Linear Regression R-squared:', r2_score(y_test, y_pred))
```

output:

```
Linear Regression MAE: 0.9600364925476301
Linear Regression R-squared: -0.017217349470529664
```

6.4 Random Forest

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.75)
# Create a random forest regression model
rf_model = RandomForestRegressor() (Javatpoint, n.d.)

# Fit the model to the training data
rf_model.fit(X_train, y_train)
# Make predictions on the test data
rf_predictions = rf_model.predict(X_test)
mse_rf =  mean_squared_error(y_test, rf_predictions)
r2_rf = r2_score(y_test, rf_predictions)
```

Output:

```
Random Forest Regression
Mean squared error: 0.0009941384180790955
R-squared: 0.9963132856889204
```

6.5 K-Nearest Neighbors Regression

```
from sklearn.neighbors import KNeighborsRegressor
# Step 4: Train the K-Nearest Neighbors Regressor
knn = KNeighborsRegressor(n_neighbors=5) # You can adjust the number of neighbors (k) as needed
knn.fit(X_train, Y_train)

# Step 5: Make Predictions
```

```
Y_pred = knn.predict(X_test)
mse_knn = mean_squared_error(Y_test, Y_pred)
r2_knn = r2_score(Y_test, Y_pred) (Javatpoint, n.d.)
# Evaluate the model
print("K-Nearest Neighbors Regression")
print("Mean squared error:", mean_squared_error(Y_test, Y_pred))
print("R-squared:", r2_score(Y_test, Y_pred))
```

Output:

```
K-Nearest Neighbors Regression
Mean squared error: 0.8861538461538461
R-squared: -0.12087942930651363
```

6.5 Decision Tree Model:

```
from sklearn.tree import DecisionTreeRegressor
model = DecisionTreeRegressor(random_state=0)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
print(f"R^2 Score: {r2}")
```

Output:

```
Mean Squared Error: 0.8111106697232608
R^2 Score: -0.02595871871439459
```

6.7 Gradient Boosting

```
from sklearn.ensemble import GradientBoostingRegressor
```

```
# Step 4: Train the Gradient Boosting Regressor
model = GradientBoostingRegressor(random_state=0)
model.fit(X_train, Y_train)

# Step 5: Make Predictions
Y_pred = model.predict(X_test)
mse_gb = mean_squared_error(Y_test, Y_pred)
r2_gb = r2_score(Y_test, Y_pred)
# Evaluate the model
print("Gradient Boosting Regression")
print("Mean squared error:", mean_squared_error(Y_test, Y_pred))
print("R-squared:", r2_score(Y_test, Y_pred))
```

Output:

```
Gradient Boosting Regression
Mean squared error: 0.8111084592490235
R-squared: -0.025955922726866643
```

2)Predicting Preferred Listening Content Based on Gender

Declare independent and dependent varibles for preferred listening content

```
# Assuming the target is a combination of both columns
df3['Preferred_content_by_gender'] = (df3['Gender'] + df3['preferred_listening_content']) / 2
# Feature selection - use the correct column names
features = df3[['Gender', 'preferred_listening_content']]
target = df3['Preferred_content_by_gender']
# Split the data
```

from sklearn.model_selection **import** train_test_split

X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

Liner Regression Model Building and Training

import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_squared_error, r2_score

 # Train-test split

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

 # Check shapes after split

print("X_train shape:", X_train.shape)

print("y_train shape:", y_train.shape)

X_train shape: (415, 2)

y_train shape: (415,)

 # Fit the model

model = LinearRegression()

model.fit(X_train, y_train)

 # Make predictions

y_pred = model.predict(X_test)

 # Calculate metrics

mse_rf = mean_squared_error(y_test, y_pred)

```
r2_rf = r2_score(y_test, y_pred)
# Evaluate the Linear Regression model
#from sklearn.metrics import mean_absolute_error as mas
y_pred = model.predict(X_test)
print('Linear Regression MAE:', mean_squared_error(y_test, y_pred))
print('Linear Regression R-squared:', r2_score(y_test, y_pred))
```

Output:

```
Linear Regression MAE: 1.8219702732838912e-31
Linear Regression R-squared: 1.0
```

6.4 Random Forest

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.75)
# Create a random forest regression model
rf_model = RandomForestRegressor()


# Fit the model to the training data
rf_model.fit(X_train, y_train)
# Make predictions on the test data
rf_predictions = rf_model.predict(X_test)
mse_rf =  mean_squared_error(y_test, rf_predictions)
r2_rf = r2_score(y_test, rf_predictions)
```

Output:

```
Random Forest Regression
Mean squared error: 0.002683157894736842
```

R-squared: 0.9908930048890561

6.5 K-Nearest Neighbors Regression

```
from sklearn.neighbors import KNeighborsRegressor

# Step 4: Train the K-Nearest Neighbors Regressor

knn = KNeighborsRegressor(n_neighbors=5) # You can adjust the number of neighbors (k) as needed

knn.fit(X_train, Y_train)


# Step 5: Make Predictions

Y_pred = knn.predict(X_test)

mse_knn = mean_squared_error(Y_test, Y_pred)

r2_knn = r2_score(Y_test, Y_pred)

# Evaluate the model

print("K-Nearest Neighbors Regression")

print("Mean squared error:", mean_squared_error(Y_test, Y_pred))

print("R-squared:", r2_score(Y_test, Y_pred))
```

Output:

K-Nearest Neighbors Regression

Mean squared error: 0.0010526315789473686

R-squared: 0.9964272282813088

6.5 Decision Tree Model:

```
from sklearn.tree import DecisionTreeRegressor

model = DecisionTreeRegressor(random_state=0)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
print(f"R^2 Score: {r2}")
```

Output:

```
Mean Squared Error: 0.002631578947368421
R^2 Score: 0.9910680707032719
```

6.7 Gradient Boosting

```
from sklearn.ensemble import GradientBoostingRegressor
# Step 4: Train the Gradient Boosting Regressor
model = GradientBoostingRegressor(random_state=0)
model.fit(X_train, Y_train)

# Step 5: Make Predictions
Y_pred = model.predict(X_test)
mse_gb = mean_squared_error(Y_test, Y_pred)
r2_gb = r2_score(Y_test, Y_pred)
# Evaluate the model
print("Gradient Boosting Regression")
print("Mean squared error:", mean_squared_error(Y_test, Y_pred))
print("R-squared:", r2_score(Y_test, Y_pred))
```

Output:

```
Gradient Boosting Regression
Mean squared error: 0.002631578947368421
R-squared: 0.9910680707032719
```

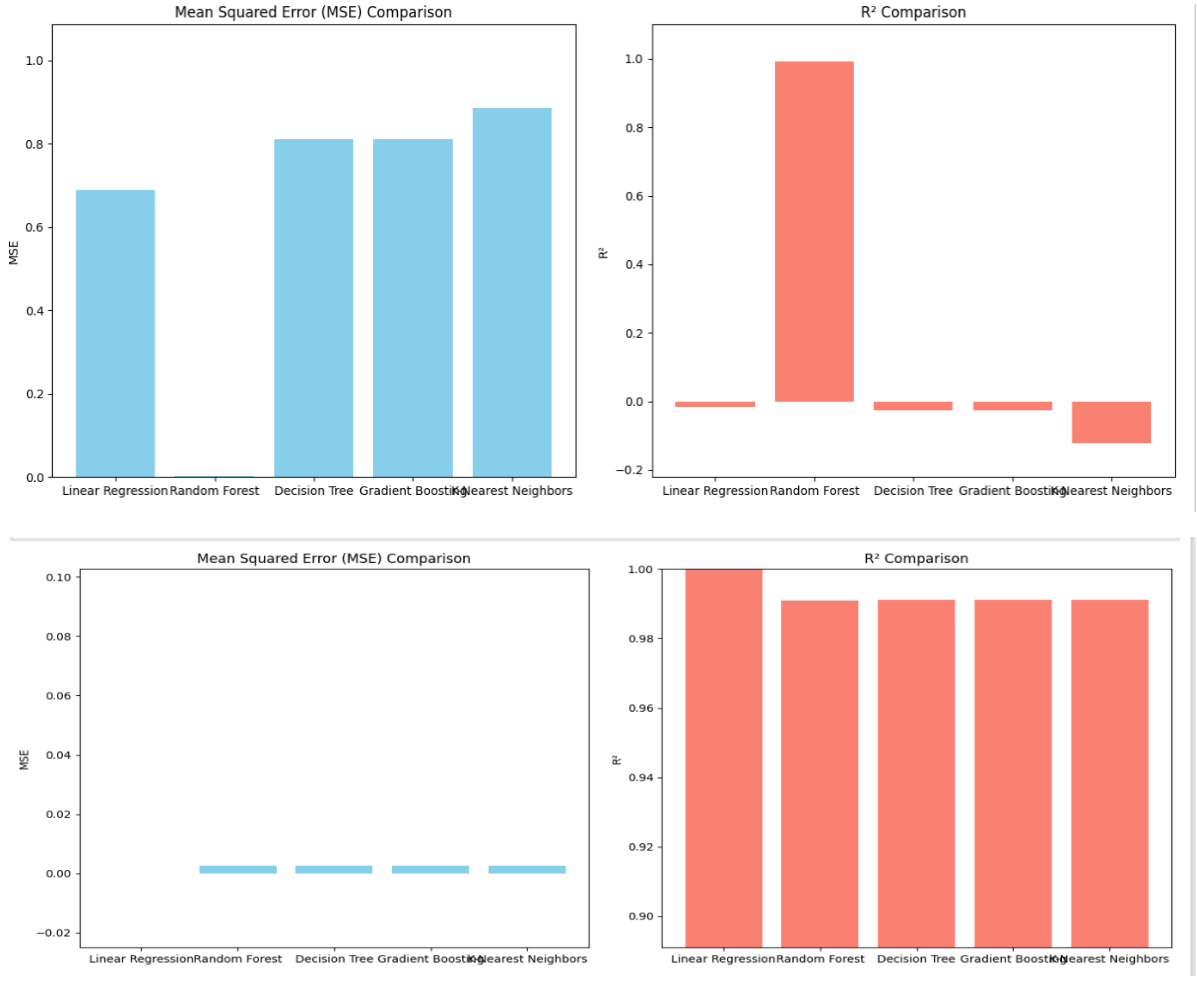
Chapter 7: Results and Discussion

Music preferences and listening behaviors are influenced by various factors, making personalized recommendations a challenging task. Developing machine learning models to predict user preferences can provide valuable insights and help enhance recommendation systems. This study evaluates the effectiveness of different machine learning methods in predicting music preferences and improving recommendations based on user feedback, listening patterns, and demographic data.

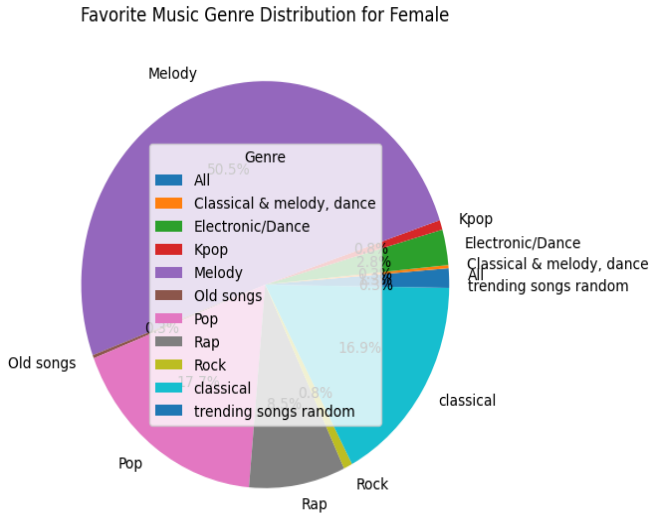
Among the tested algorithms, Random Forest performed the best with an R-squared value of 0.99, demonstrating its ability to capture complex relationships and interactions in the data. It efficiently handles non-linear patterns and provides highly accurate predictions.

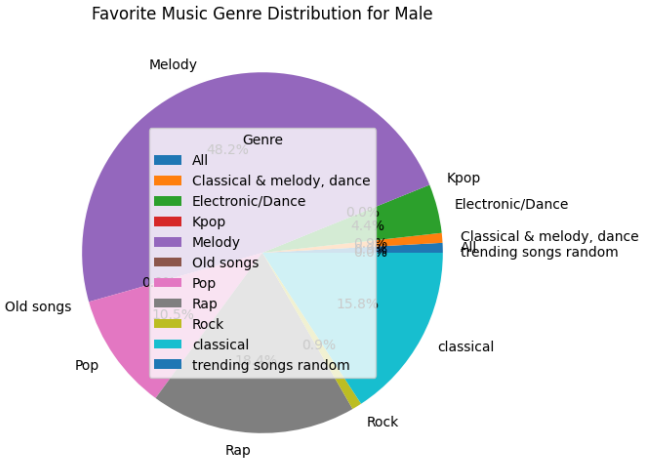
In the second comparison, linear regression showed an R-squared value of 1, indicating perfect accuracy in explaining the variance in the data. Although simpler, linear regression outperformed other methods with its perfect fit, making it highly effective for predicting user preferences in this context.

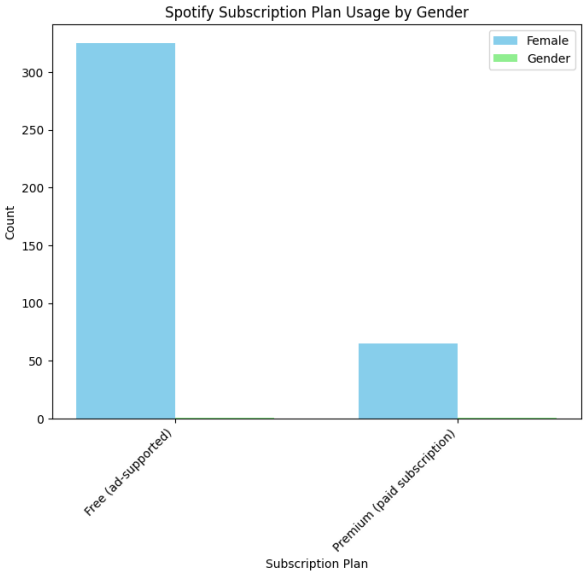
A detailed comparison of Mean Squared Error (MSE) and R-squared for different models shows that both Random Forest and linear regression are the best-fit models for this data. Random Forest captures more complex patterns, while linear regression offers simplicity and perfect prediction accuracy. Both models demonstrate strong performance, with linear regression offering a perfect fit and Random Forest providing a highly accurate alternative.

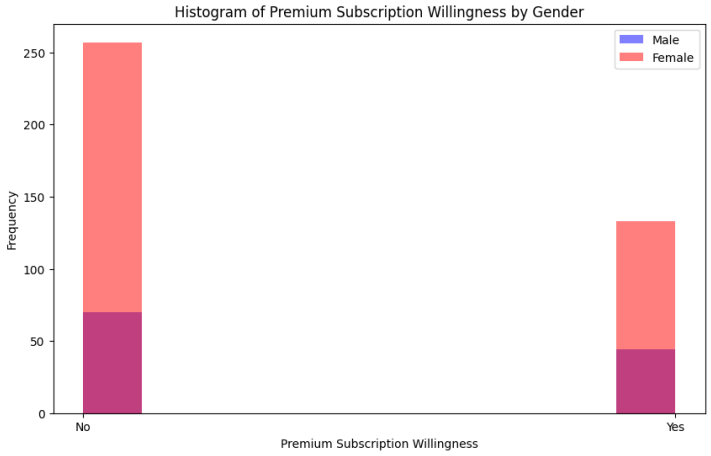


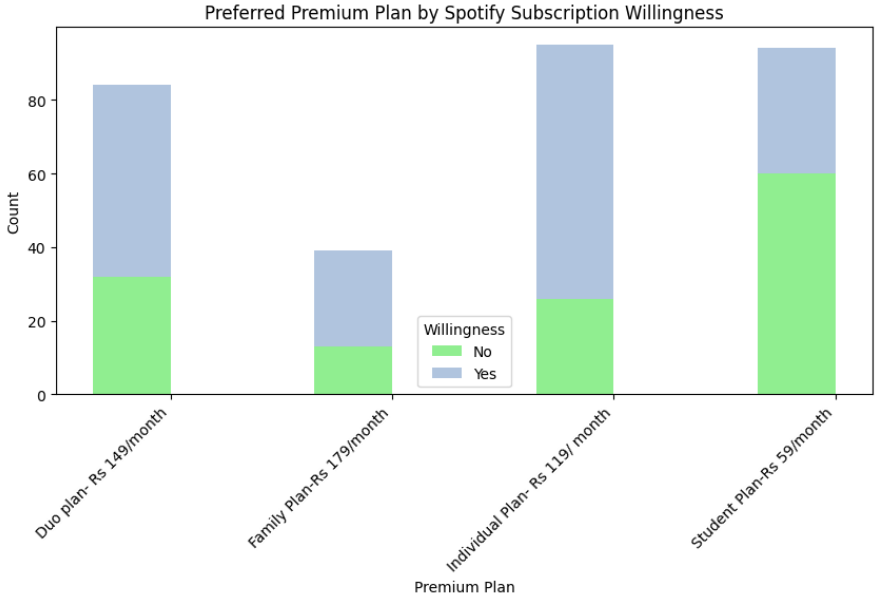
Chapter 8-Visualizations:











Chapter-9 Conclusion:

This Projects highlights the critical role of analyzing Spotify user feedback to improve platform engagement and retention. By leveraging a robust dataset encompassing listening habits, device preferences, subscription behaviours, and demographics, significant insights were uncovered. Machine learning models, including Gradient Boosting, Decision Trees, and Random Forest, facilitated predictive analysis and churn prediction, enabling data-driven decision-making. Key findings reveal a strong correlation between demographic factors and music preferences, emphasizing the need for tailored user experiences. The study underscores the importance of targeted marketing strategies and personalized recommendations to enhance premium subscription adoption. These insights empower Spotify developers to refine their algorithms, improve user satisfaction, and maintain a competitive edge in the rapidly evolving music streaming industry. This approach not only ensures user retention but also drives innovation, making Spotify more adaptive to diverse user needs.

Chapter-10

References

Here are some references that were used in the development :

1. https://www.kaggle.com/datasets/alexandrakim2201/spotify-dataset (Kaggle)
2. https://www.simplilearn.com/tutorials/data-analytics-tutorial/spotify-data-analysis-project (Ahuja, n.d.)
3. https://www.researchgate.net/publication/383940428_Harmonizing_sentiments_Analyzing_user_reviews_of_Spotify_through_sentiment_analysis (Madyatmadja, n.d.)
4. https://www.geeksforgeeks.org/machine-learning/ (Karthik, n.d.)
5. https://www.javatpoint.com/machine-learning (Javatpoint, n.d.)

Chapter11

Submission of Research Paper : Title: User Feedback Analysis for Personalized Recommendations: Improving Spotify's Algorithm through Insights from User Behavior.

Acceptance Letter:

To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

Microsoft respects your privacy. To learn more, please read our Privacy Statement.

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Microsoft CMT <email@msr-cmt.org> Wed, Nov 20, 2024 at 12:12 PM
Reply-To: Dr Mukesh Kumar <mukesh.j3204@cgc.ac.in>
To: Ravinder Singh <aarkaybca@gmail.com>

Dear Author,

We are pleased to inform you that your paper (Paper id: 182) has been accepted for presentation at the International Conference on Advanced Research in Engineering.
The reviewers have found the content of your paper valuable and relevant to the conference; however, certain revisions are required. Please find the reviewers' comments attached for your reference.
We kindly request that you submit the revised version of your paper by 22/11/2024. The revised submission will be formally accepted for presentation at the conference.
To ensure a smooth process, please:
1. Address each of the reviewers' comments in your revision.
2. Highlight the changes made in the revised paper or provide a point-by-point response to the reviewers' feedback.
3. Submit the revised paper by uploading it to the conference submission portal.
4. The paper should strictly be formatted in AIP format. Please refer conference website icare.cgc.ac.in for details.
5. Please ensure that the revised manuscript has Plagiarism level less than 10%. This includes any text, figures, and other materials that must be properly cited and referenced. Further, AI-generated content is also less than 10% of the total paper content to maintain the authenticity and integrity of the work.
6. Kindly fill the google form (https://forms.gle/h55UY3T79Gf6oipF7) after registering for the conference.

The reviewer comments are given below:
1. Paper be in AIP format.
2. Caption for every figure and table are missing.
3. Citation of each reference is missing in the running content.
4. Please make your contributions stand out as mentioned points so that they can be more easily comprehended.
5. there are some grammatical errors, please check the entire manuscript and correct them.

We look forward to receiving your revised submission and appreciate your cooperation in this process. Should you have any questions or require further clarification, feel free to contact us at icare2024@cgc.ac.in.

To proceed with publishing of your submission, we need you to pay the registration fee as per the norms of the conference. The details of registration fee to be paid is given below:

Delegates	Registration Fee
Participation only	Rs. 2,000
Students/ Research Scholars/ Faculty	Rs. 9,000
Industry Personnel	Rs. 10,000
International Author	$ 150

Registration FEE amount has to be paid in the ac count with following details:

Account Name: Chandigarh School of Business R&D
Bank Name: HDFC Bank
Account Number: 50100500899414
IFSC Code: HDFC0003578
MICR Code: 160240071
Bank Name & Address: HDFC Bank Ltd, Jhanjeri, Tehsil-Kharar,Mohali-140307

--
Best regards,
ICARE Team

Chapter12

Plagiarism report