### **Business Understanding Report: Spotify Song Recommendations Project**

Group D11

Ron-Aaron Vahtra, Reno Feliks Lindvere, Martin Koltsov

**Background**The rapid growth of music streaming services like Spotify has made personalized recommendations a key differentiator in user experience. This project aims to leverage data from Spotify to build a recommendation system based on user input (song(s) or playlist) and predict the potential success of songs using their audio features.

**Business Goals**

1. Enhance user engagement by providing personalized song recommendations based on preferences.
2. Support music curators, marketers, and artists by identifying potential hits based on audio characteristics.
3. Deliver actionable insights for improving playlist design and music promotion strategies.

**Business Success Criteria**

* **Quantitative**: Achieve at least 80% user satisfaction in a simulated test environment where users rate recommendations.
* **Qualitative**: Develop a model that successfully identifies songs with a popularity score above a threshold of 75 (on Spotify's scale) at least 85% of the time.

**Inventory of Resources**

* **People**: Project lead, data analyst, ML engineer, domain expert (music industry).
* **Data**:
  + Dataset 1: Spotify tracks dataset from Kaggle (~20.12 MB).
  + Dataset 2: Spotify API datasets, including audio features and analysis.
* **Tools**: Python libraries (Pandas, NumPy, Scikit-learn, TensorFlow), data visualization tools (Matplotlib, Seaborn), Spotify API for real-time data fetching.
* **Hardware**: Personal computers with sufficient computational power or cloud platforms (e.g., Google Colab).
* **Other**: Spotify Developer account for API access.

**Requirements, Assumptions, and Constraints**

* **Requirements**:
  + Access to Spotify API data.
  + Adherence to Spotify's API usage limits and terms of service.
* **Assumptions**:
  + Audio features correlate strongly with user preferences and song popularity.
* **Constraints**:
  + Limited labeled data for predicting hits.
  + API rate limits affecting data-fetching speeds.

**Risks and Contingencies**

* **Risk**: Inadequate API quota may delay data fetching.
  + **Contingency**: Pre-fetch and cache data for commonly queried songs/playlists.
* **Risk**: Model underperformance due to lack of labeled data for hit prediction.
  + **Contingency**: Use transfer learning with existing datasets or augment data with synthetic samples.

**Terminology**

* **Danceability**: A measure describing how suitable a track is for dancing.
* **Popularity**: A score (0-100) provided by Spotify, reflecting song popularity.
* **Audio Features**: Characteristics like tempo, energy, and loudness.

**Costs and Benefits**

* **Costs**: Developer time (200 hours), Spotify API usage (free tier), computational resources ($50 for cloud computing).
* **Benefits**: Increased user engagement for music streaming platforms, better playlist curation tools for users, and actionable insights for artists and marketers.

**Data-Mining Goals**

1. Develop a recommendation model that predicts user preferences based on input tracks or playlists.
2. Build a classification model to predict whether a song will achieve a popularity score above 75.

**Data-Mining Success Criteria**

* Recommendation system with a Top-10 recommendation accuracy of at least 80%.
* Classification model with a prediction accuracy of at least 85% on the test set.

**Project Plan**

1. **Data Understanding**
   * Explore datasets from Kaggle and Spotify API to assess feature quality.
   * Visualize relationships between audio features and popularity.
2. **Data Preparation**
   * Preprocess datasets (handle missing values, normalize features).
   * Merge user-provided playlist data with Spotify datasets for model training.
3. **Modeling**
   * Recommendation: Use collaborative filtering and content-based filtering techniques.
   * Hit prediction: Train classifiers like Logistic Regression, Random Forest, and Neural Networks.
4. **Evaluation**
   * Validate recommendation model using user feedback simulations.
   * Evaluate hit prediction models using F1 score and ROC-AUC metrics.
5. **Deployment**
   * Build a user-friendly interface for inputting songs or playlists and displaying recommendations.

**Initial Assessment of Tools and Techniques**

* Use Spotify's API for live data fetching and Python-based ML frameworks for modeling. Scikit-learn will suffice for initial prototyping, with TensorFlow/Keras for deep learning tasks. Data visualization tools like Seaborn and Matplotlib will be essential for insights.