# **Spotify Sentiment Analysis: Research Paper**

## **Introduction**

### **What is Sentiment Analysis?**

Sentiment analysis, or opinion mining, is a Natural Language Processing (NLP) technique that identifies, extracts, and classifies emotional tone within a text. It is widely used to gauge public opinion, detect customer satisfaction, and track psychological patterns. Text data is analyzed to determine polarity (positive, negative, neutral) and in some cases, subjectivity.

In the music industry, sentiment analysis can be applied to lyrics, user feedback, and playlist descriptions to uncover how people emotionally relate to songs. Spotify, as a leading music platform, offers a wealth of user-generated and platform-curated content that reflects listeners' emotions and preferences.

### **Project Overview**

This project explored sentiment analysis on Spotify-related data by implementing lexicon-based methods (VADER and TextBlob) and a supervised machine learning model (Logistic Regression). It also incorporated data visualization techniques and built a Streamlit application to allow users to interact with the sentiment analysis system in real time.

We used three datasets:

* A lyrics dataset containing song lyrics
* A playlist dataset with titles and descriptions
* A user behavior survey dataset reflecting listening habits, moods, and preferences

The goal was to preprocess these datasets, analyze sentiment using different models, compare the results, build an ML classifier, and present it all through visualizations and a user interface.

### **Key Contributions**

* Implemented VADER and TextBlob sentiment analysis on lyrics and playlist text.
* Analyzed user behavior data and correlated it with music sentiment trends.
* Built and trained a Logistic Regression model with TF-IDF features.
* Designed multiple data visualizations (bar charts, heatmaps, boxplots).
* Developed an interactive Streamlit app for real-time sentiment prediction.

## **Literature Review of Sentiment Analysis Models**

### **Lexicon-Based Approaches**

1. **VADER (Valence Aware Dictionary and sEntiment Reasoner)**
   * Designed for social media and informal text
   * Considers punctuation, capitalization, and degree modifiers
   * Returns compound score between -1 (negative) and +1 (positive)
   * Best suited for short-form content like song lyrics and playlist titles
2. **TextBlob**
   * Calculates two metrics: polarity (-1 to +1) and subjectivity (0 to 1)
   * Simpler than VADER, often used for quick sentiment analysis
   * Handles moderately structured content well

### **Machine Learning Approaches**

* Use feature extraction (TF-IDF) and supervised models (e.g., Logistic Regression)
* Offer adaptability and improve with more labeled training data
* Require preprocessing, feature engineering, model training, evaluation

### **Related Work**

* **Movie Reviews**: Naive Bayes, Logistic Regression (~85% accuracy)
* **Twitter**: Real-time public sentiment using VADER
* **Music Mood Classification**: LSTM used on audio and lyrics

Our project adapts and expands on these methodologies by combining them into a hybrid analytical pipeline specifically focused on Spotify data.

## **Dataset Collection & Description**

### **1. Lyrics Dataset**

* File: lyrics.csv
* Sourced from Kaggle
* Contains artist, song, link, and text
* Used for sentiment extraction and ML training

### **2. Playlist Dataset**

* File: playlists\_from\_json.csv
* Extracted using Spotify’s public API and processed from JSON
* Fields: title, description
* Used to analyze user-defined playlist emotions

### **3. User Behavior Dataset**

* File: user\_behaviour.xlsx
* Survey with 20+ fields: age, gender, genre preferences, device, mood influence
* Cleaned and summarized to analyze listening context and mood correlation

### **Preprocessing Pipeline**

* Dropped nulls, standardized text
* Tokenized and lowercased text entries
* Combined the datasets into combined\_sentiment.csv
* Saved clean versions into the /processed/ folder

We created preview samples to confirm that data was correctly aligned across files. Lyrics had approximately 50,000 rows, playlists had around 10,000 entries, and user behavior had over 1,000 rows.

## **Methodology & Implementation**

### **Step 1: VADER Sentiment Analysis**

* Applied VADER compound scoring to all text fields
* Converted compound scores to labels: Positive, Neutral, Negative
* Results stored in lyrics\_sentiment.csv and playlists\_sentiment.csv

### **Step 2: TextBlob Analysis**

* Calculated polarity and subjectivity for each entry
* Created new columns: textblob\_polarity, textblob\_sentiment
* Output stored in textblob\_sentiment.csv

### **Step 3: Machine Learning Model**

* Used VADER labels as targets for supervised model
* Extracted features using TF-IDF vectorizer (max 5000 features)
* Trained Logistic Regression classifier
* Accuracy on test set: ~83%
* Model and vectorizer saved using joblib

### **Folder Structure**

* /scripts/: All Python files
* /models/: Saved ML models
* /results/: CSV outputs for sentiment scores
* /visuals/: Saved charts (PNG)
* /main\_app.py: Streamlit interface for sentiment analysis

## **Sentiment Analysis Results**

### **VADER Results**

* Dominant class: **Positive** (over 60%)
* Lyrics showed more variation across sentiment categories
* Playlist descriptions tended to be neutral or mildly positive

#### **Visualizations:**

* sentiment\_overall.png: Global sentiment bar chart
* sentiment\_by\_source.png: Split by dataset type
* sentiment\_pie\_chart.png: Circular ratio chart
* sentiment\_score\_boxplot.png: Distribution of compound scores

### **TextBlob Results**

* Produced similar polarity distribution to VADER
* Subjectivity scores highlighted personal vs factual nature

#### **Visualizations:**

* textblob\_sentiment\_distribution.png
* textblob\_sentiment\_by\_source.png
* textblob\_score\_boxplot.png

## **User Behavior Insights**

### **Visual Summary**

* user\_mood\_influence.png: Most influential factors for using Spotify
* fav\_genres.png: Most popular genres by user count

### **Observations:**

* Users between ages 20–35 made up the majority
* Most users prefer music to relax, study, or uplift mood
* Pop, Lo-fi, and Indie were top-rated genres
* Peak usage during mornings and late evenings

### **Correlation with Sentiment**

* Higher subjectivity observed in responses from users preferring mood-based playlists
* Genre-specific playlists tended to align with positive sentiments
* Frequency of listening positively correlated with sentiment polarity

## **Streamlit Web Application**

### **App Features Implemented**

* Input box for single-text prediction
* CSV upload for batch analysis
* Drop-down to choose sentiment analyzer: VADER, TextBlob, or ML Model
* Display of predicted label + probability
* Dynamic bar chart of sentiment distribution

### **Tools Used**

* Streamlit (for UI)
* Pandas (data handling)
* Joblib (model deployment)
* Matplotlib/Seaborn (visuals)

### **Launch Instructions**

streamlit run main\_app.py

The app was tested with both individual sentences and full CSV datasets, confirming end-to-end functionality.

## **Conclusion & Future Scope**

### **Project Achievements**

* Combined structured (survey) and unstructured (text) data
* Cleaned and preprocessed over 60,000 entries
* Applied three sentiment analysis models
* Built visual summaries for effective reporting
* Delivered a usable and intuitive Streamlit app

### **Learnings**

* VADER excels at short, informal text sentiment
* TextBlob provides complementary analysis with subjectivity
* Logistic Regression with TF-IDF is effective for domain-specific classification
* Streamlit bridges model results with user accessibility

### **Future Enhancements**

* Expand to real-time Spotify track analysis using Spotify Web API
* Support multilingual sentiment analysis for global users
* Incorporate audio features (tempo, mode) for hybrid modeling
* Upgrade from logistic regression to deep learning models (BERT, BiLSTM)

### **Deliverables**

* CSVs: combined\_sentiment.csv, textblob\_sentiment.csv, etc.
* PNGs: 13+ sentiment and user-based graphs
* Python Scripts: Preprocessing, analysis, model training
* Streamlit App: main\_app.py

**End of Research Paper**