An Analytics Approach for Kalshi Market Insights

Predicting the #1 Song on Spotify USA

Prepared by Kenneth Lent

MDA 720 Applied Business Analytics
Capstone
May 12, 2025

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY
2. BACKGROUND
   2.1. Introduction to the Problem/Opportunity
   2.2. The "Streaming-Sentiment Signals" Analytical Tool Concept
   2.3. Project Necessity & Market Relevance
3. OBJECTIVE/GOALS OF THE PROJECT
   3.1. Primary Objective
   3.2. Specific Goals
4. DATA EXTRACTION/COLLECTION
   4.1. Data Sources & Rationale
   4.1.1. Spotify Charts CSVs
   4.1.2. Spotify Web API
   4.1.3. Attempted: Google Trends (Pytrends)
   4.2. Data Storage
5. DATA PREPROCESSING AND FEATURE ENGINEERING
   5.1. Data Cleaning and Transformation
   5.2. Feature Engineering
6. DATA EXPLORATION AND VISUALIZATION
   6.1. Dataset Overview
   6.2. Feature Correlation with Target
   6.3. Distribution of Predicted Probabilities
7. PREDICTIVE MODELING AND ANALYSIS
   7.1. Model Selection: LightGBM
   7.2. Training and Testing Strategy: Time-Based Rolling Window
   7.3. Model Evaluation
   7.3.1. Multi-Day Performance Metrics
   7.3.2. ROC AUC and Precision-Recall
   7.3.3. Calibration
   7.3.4. Cumulative Hit Rate
8. RESULTS AND DISCUSSION
   8.1. Model Performance Summary
   8.2. Key Feature Insights
   8.3. Implications for Kalshi Market Trading

9. CONCLUSIONS AND RECOMMENDATIONS
   9.1. Conclusion
   9.2. Recommendations for the Analytical Tool
   9.3. Limitations
   9.4. Future Work
10. BIBLIOGRAPHY/REFERENCES

---

1. EXECUTIVE SUMMARY

This project presents the development of "Streaming-Sentiment Signals," an analytical tool designed to provide a competitive edge in Kalshi's prediction market for "Which song will be #1 on Spotify USA tomorrow?". By ingesting historical Spotify chart data and augmenting it with track metadata via the Spotify Web API, this project engineered relevant features reflecting a song's momentum and intrinsic characteristics. A LightGBM classification model was trained using a time-based rolling window approach to forecast the probability of a song reaching the #1 spot.

Over a multi-day evaluation period (January 1, 2025, to May 11, 2025), the model demonstrated strong predictive capability, achieving an overall ROC AUC score of 0.97. On average, when predicting the #1 song, the model achieved a precision of 0.68 and a recall of 0.68. Key features influencing the predictions included current rank, stream count, and stream momentum. While these results are promising for informing trading strategies, direct profitability requires backtesting against actual market odds. The project highlights the value of timely data collection and robust feature engineering, laying a foundation for a potentially monetizable insights service.

---

2. BACKGROUND

2.1. Introduction to the Problem/Opportunity

Kalshi offers event contracts on a variety of real-world outcomes, including daily financial markets on popular culture phenomena. One such market is "Which song will be #1 on Spotify USA tomorrow?". This market settles daily at 11 a.m. ET based on Spotify's official U.S. Top 50 chart for the previous day. Participants buy "Yes" or "No" contracts for specific songs, with prices (ranging from 1¢ to 99¢) reflecting the market's collective belief in the probability of that song reaching the #1 position.

2.2. The "Streaming-Sentiment Signals" Analytical Tool Concept

The core idea is to develop an analytical tool, "Streaming-Sentiment Signals," that

leverages publicly available data related to song popularity and momentum. These signals, such as streaming counts, chart movements, and potentially search interest or lyric engagement, often become available or show significant changes *before* the Kalshi market settles. A data-driven model that can process these signals and generate calibrated probabilities for each contending song can identify mispricings in the Kalshi market, offering a potential edge to traders.

2.3. Project Necessity & Market Relevance

The Kalshi market for Spotify's #1 song is dynamic, influenced by new releases, viral trends, and promotional activities. Traders relying solely on intuition or lagging indicators may miss opportunities. An analytical tool that systematically ingests and models leading indicators can provide more objective, timely, and potentially profitable trading signals. This project aims to build the foundational predictive model for such a tool.

---

3. OBJECTIVE/GOALS OF THE PROJECT

3.1. Primary Objective

To develop and evaluate a machine learning model that predicts the probability of a song reaching the #1 spot on the Spotify USA daily chart, thereby providing actionable insights for Kalshi market participants.

3.2. Specific Goals

1. **Data Ingestion:** Collect historical Spotify daily chart data and relevant track metadata using available APIs.
2. **Feature Engineering:** Create features that capture song momentum (e.g., changes in rank, stream velocity) and track characteristics (e.g., popularity, explicitness, days since release).
3. **Predictive Modeling:** Train a classification model to forecast the #1 song based on the engineered features.
4. **Model Evaluation:** Assess the model's performance using appropriate metrics like ROC AUC, precision, recall, and calibration over a historical period.
5. **Output Generation:** Create an output (e.g., a daily report or data feed) that translates model predictions into potentially actionable trading information.

---

4. DATA EXTRACTION/COLLECTION

4.1. Data Sources & Rationale

- **4.1.1. Spotify Charts CSVs:**
    - **Source:** Historical daily chart data (e.g., regional-us-daily-YYYY-MM-DD.csv) typically available from Spotify or chart aggregators. For this project, these were assumed to be locally stored.
    - **Content:** Daily rank, track URI, artist names, track name, streams, peak rank, previous rank, days on chart.
    - **Rationale:** Provides the core historical performance data and the ground truth (target variable) for the #1 song.
- **4.1.2. Spotify Web API:**
    - **Source:** Official Spotify Web API (accessed using the spotipy library).
    - **Content:** Track-specific metadata such as popularity score, duration_ms, explicit status, and release_date. This data was fetched in batches for unique track URIs and cached locally in track_meta_cache.json to manage API rate limits and speed up subsequent runs.
    - **Rationale:** Augments chart data with intrinsic track features that might influence popularity and chart performance. API usage is a key technique covered in the course.
- **4.1.3. Attempted: Google Trends (Pytrends):**
    - An attempt was made to incorporate Google Trends data for track and artist search interest using the pytrends library, as proposed initially. However, due to aggressive rate limiting by Google Trends, collecting this data at a daily granularity for a dynamic list of songs proved unfeasible within the project's scope and resources. This highlights a real-world challenge in relying on certain public data sources for high-frequency signals.

4.2. Data Storage

The raw daily chart data was stored as individual CSV files. The Spotify API metadata was cached in a JSON file (track_meta_cache.json). During processing, data was loaded into pandas DataFrames. The final multi-day predictions were saved to an Excel file (multi_day_predictions_2025-01-01_to_05-11.xlsx).

5. DATA PREPROCESSING AND FEATURE ENGINEERING

5.1. Data Cleaning and Transformation

- **Date Handling:** Chart dates were robustly extracted from CSV filenames and converted to datetime objects. Release dates from the Spotify API were also converted to datetime objects, with errors coerced to NaT.
- **Merging:** The daily chart data was merged with the Spotify API metadata based on the track URI.
- **Data Type Conversion:** explicit status was converted to an integer (0 or 1).

5.2. Feature Engineering

To capture the dynamics of song performance, the following features were engineered (Cell 6 of the notebook):

- days_since_release: Calculated as (chart_date - release_date).
- rank_prev: The song's rank on the previous day (using groupby('uri')['rank'].shift(1)).
- rank_change: The change in rank from the previous day (rank_prev - rank).
- streams_roll3: A 3-day rolling mean of streams for each song.
- stream_momentum: The difference between current day's streams and the 3-day rolling mean (streams - streams_roll3).

The candidate universe for prediction on any given day was limited to the top N=10 tracks from the previous day's chart to focus computational resources on the most likely contenders. The target variable (target) was a binary indicator (1 if the song was #1 on the *next* day, 0 otherwise).

The final feature set used for modeling included: rank, streams, rank_change, stream_momentum, popularity, duration_ms, explicit, and days_since_release.

6. DATA EXPLORATION AND VISUALIZATION

6.1. Dataset Overview

The feature engineering process resulted in a dataset of 3,750 observations (song-day instances) for model training and evaluation, covering 375 unique chart dates. Among these, there were 359 instances where a song became #1 the next day (positive examples).

6.2. Feature Correlation with Target

A correlation analysis was performed to understand the linear relationship between the engineered features and the target variable (predicting the next-day #1 song) across all available dates.

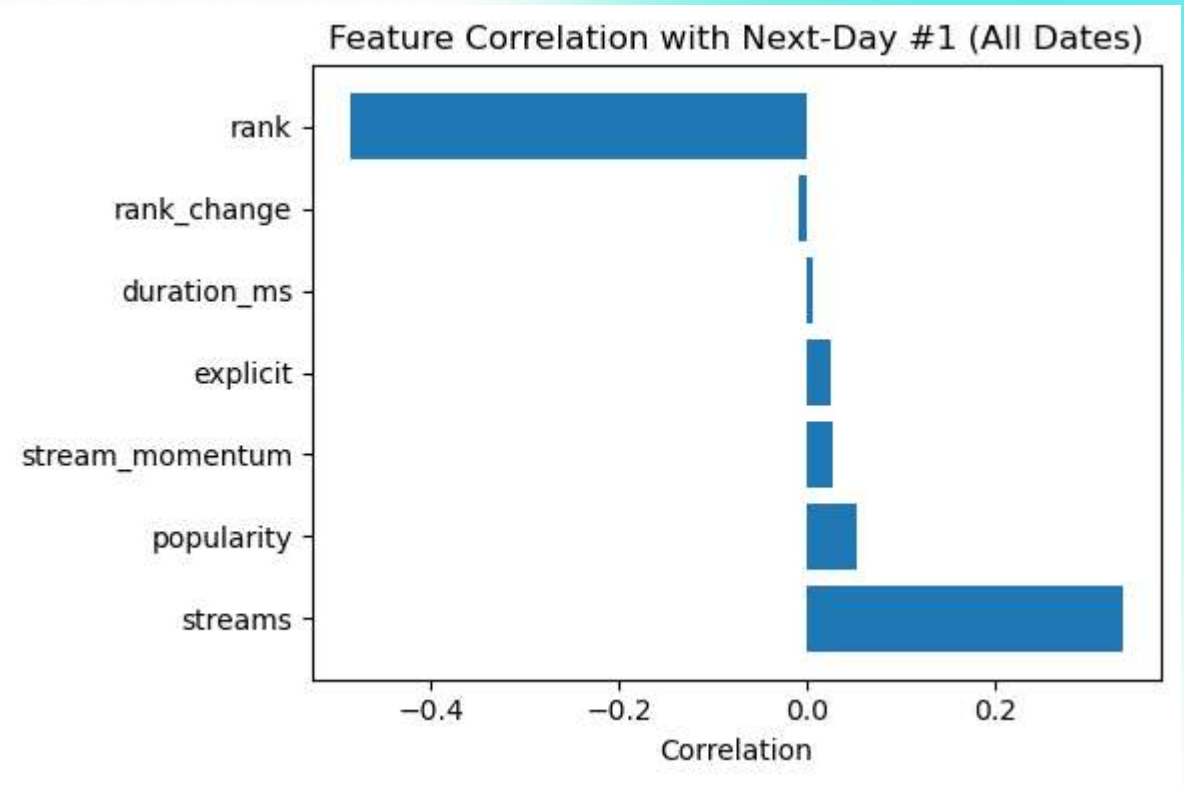

Figure 1: Feature Correlation with Next-Day #1

The plot indicates that:

- streams: Shows a strong positive correlation, suggesting higher stream counts are associated with a higher likelihood of being #1.
- rank: Shows a strong negative correlation, as expected (lower rank number means closer to #1).
- stream_momentum, popularity, and explicit status show moderate positive correlations.
- duration_ms and rank_change show weaker correlations.

6.3. Distribution of Predicted Probabilities

The histogram of predicted probabilities from the multi-day evaluation period shows the model's confidence levels.

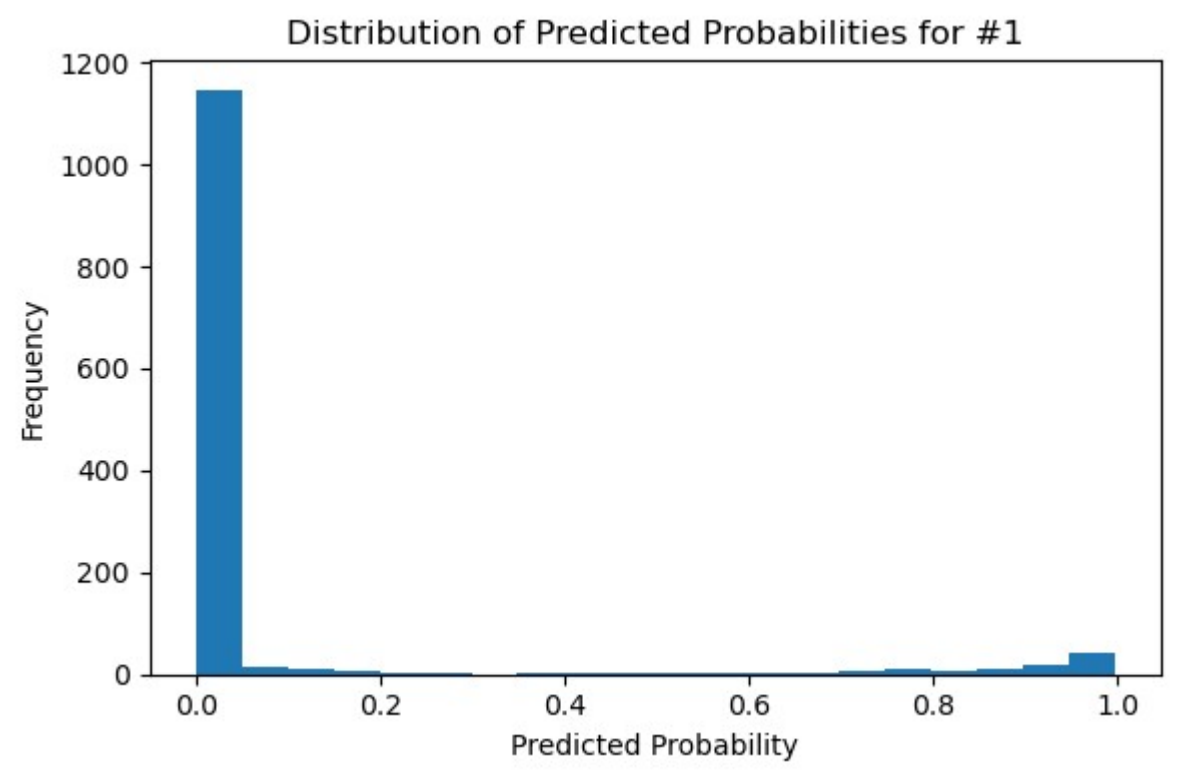

Figure 2: Distribution of Model's Predicted Probabilities for #1 Song

The distribution is heavily skewed towards low probabilities, which is expected given the rarity of a song being #1. However, the model does assign higher probabilities to a subset of songs, indicating its ability to differentiate potential #1 hits.

---

7. PREDICTIVE MODELING AND ANALYSIS

7.1. Model Selection: LightGBM

A Light Gradient Boosting Machine (LightGBM) classifier was chosen for its efficiency, ability to handle categorical features (though primarily numerical features were used here), and strong performance in many classification tasks. Key parameters included a binary objective, binary_logloss metric, learning rate of 0.05, and class_weight='balanced' to handle the imbalance between #1 and non-#1 songs.

7.2. Training and Testing Strategy: Time-Based Rolling Window

To simulate a real-world scenario where predictions are made for the next day based on historical data, a multi-day evaluation was conducted (Cell 8 of the notebook). For each day in the evaluation window (January 1, 2025, to May 11, 2025):

1. The model was trained on all data *prior* to that day.
2. Predictions were made for the songs on that specific day (i.e., predicting if they would be #1 on the *following* day).
   This rolling window approach ensures that the model is always trained on past data and tested on unseen future data, providing a more realistic assessment of its performance.

7.3. Model Evaluation

- **7.3.1. Multi-Day Performance Metrics**
  Daily precision, recall, and F1-score for predicting the #1 song (class '1') were calculated over the 130-day evaluation period.

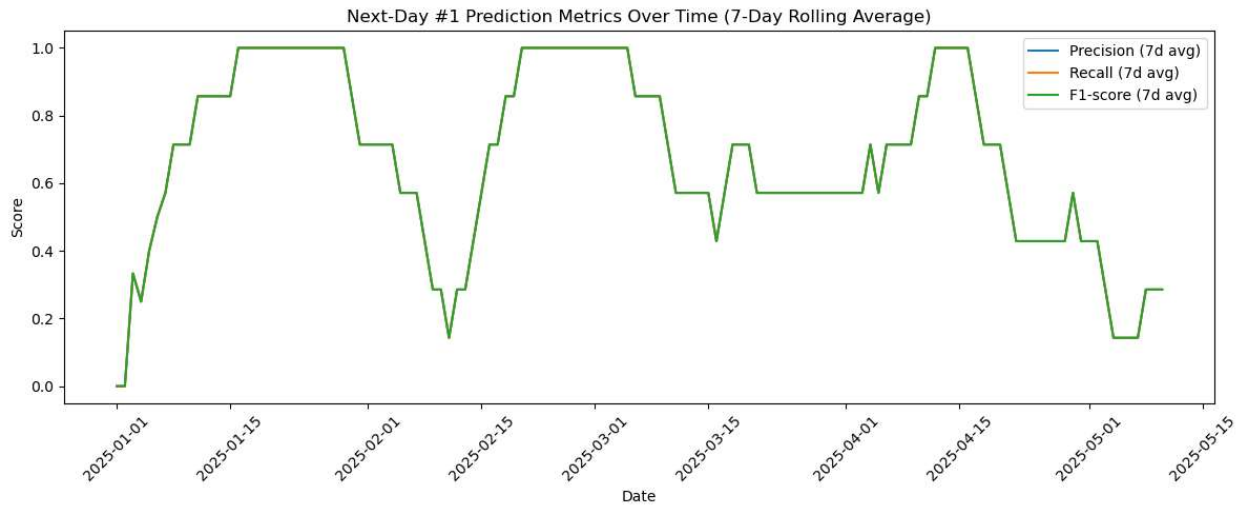

Figure 3: Next-Day #1 Prediction Metrics Over Time (Precision, Recall, F1-score for Class 1)

The metrics fluctuate daily, which is expected in a dynamic market. On average, the model achieved (from Cell 246 output):

- Mean Precision (for #1 song): 0.68
- Mean Recall (for #1 song): 0.68

- **7.3.2. ROC AUC and Precision-Recall**
  Aggregating all predictions from the multi-day evaluation:
  - The overall ROC AUC score was 0.97, indicating excellent discriminative power between songs that will become #1 and those that will not.

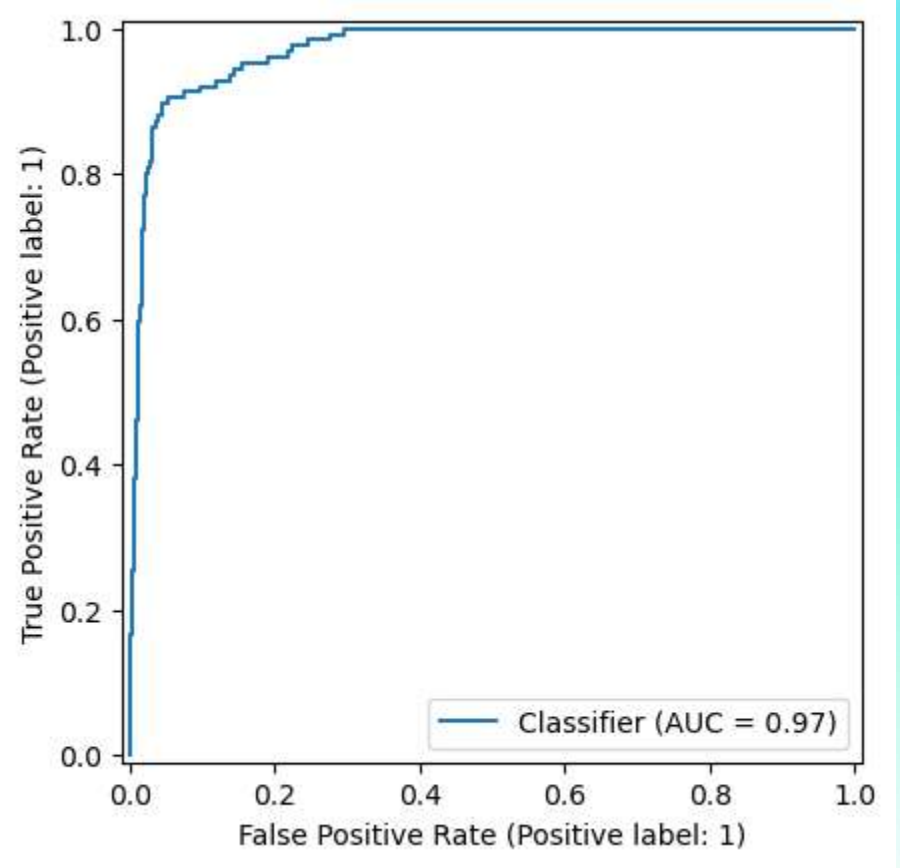

Figure 4: Receiver Operating Characteristic (ROC) Curve (All Dates)

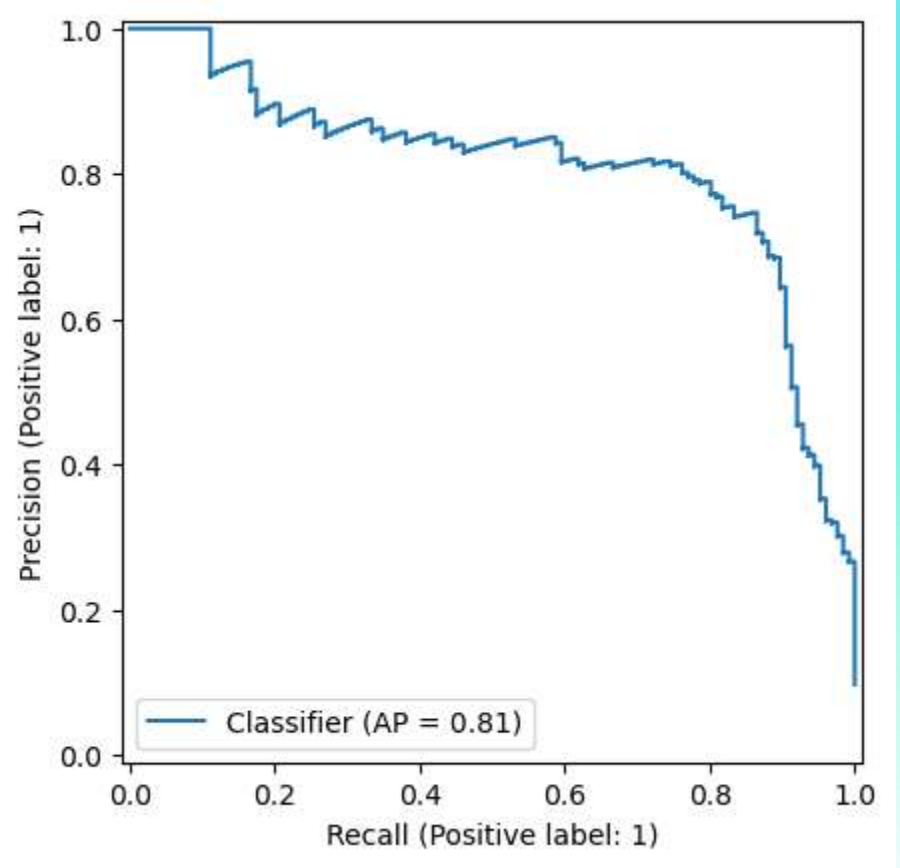

Figure 5: Precision-Recall (PR) Curve (All Dates)
The Average Precision (AP) of 0.81 on the PR curve further supports the model's ability to identify true #1 songs effectively, especially important in imbalanced datasets.

- **7.3.3. Calibration**
  The calibration curve assesses whether the predicted probabilities are reliable (e.g., if the model predicts a 70% chance, does it happen 70% of the time?).

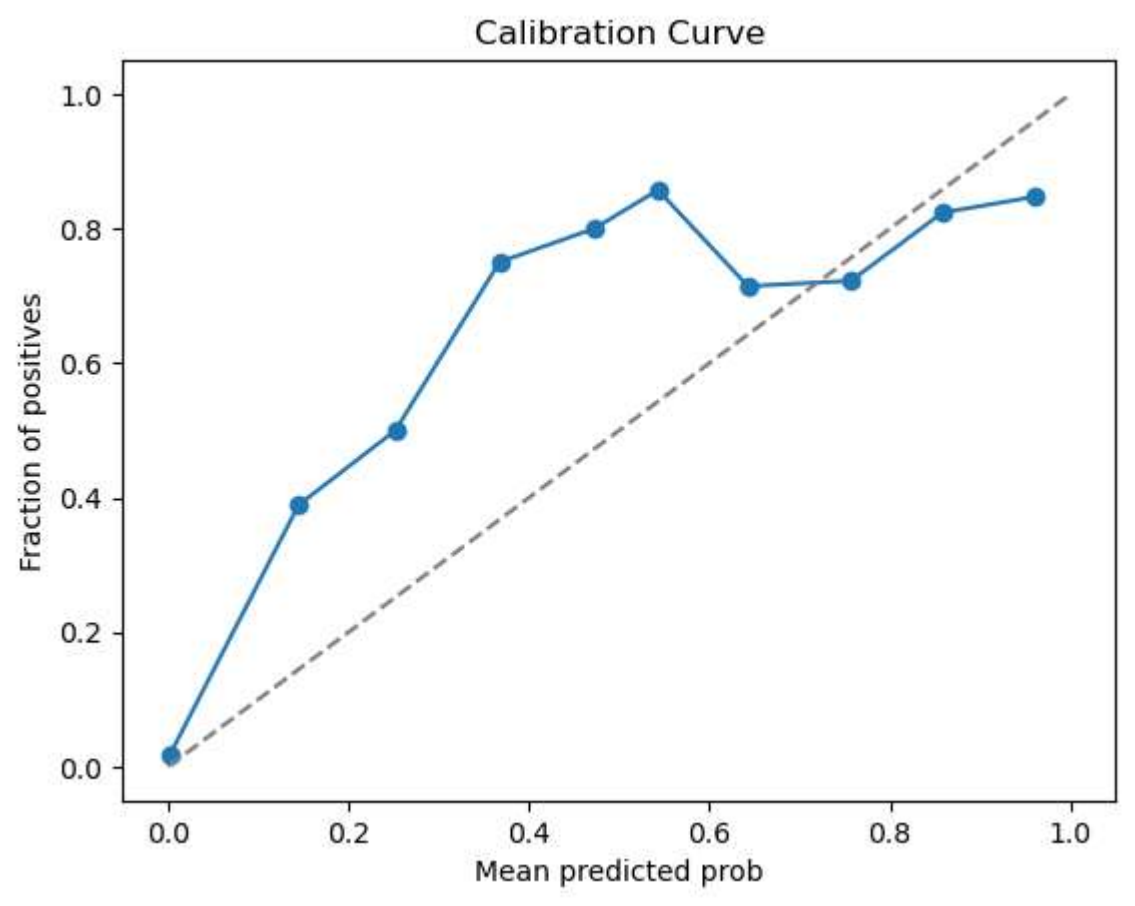

Figure 6: Calibration Curve (All Dates)
The calibration curve shows that the model's probabilities are reasonably well-calibrated, though with some deviations, particularly at higher predicted probabilities. This suggests that while the model is good at ranking songs, further calibration might refine the direct interpretation of its probabilities as true likelihoods.

- **7.3.4. Cumulative Hit Rate**
  This metric tracks whether the model correctly identified the #1 song on any given day throughout the evaluation period.

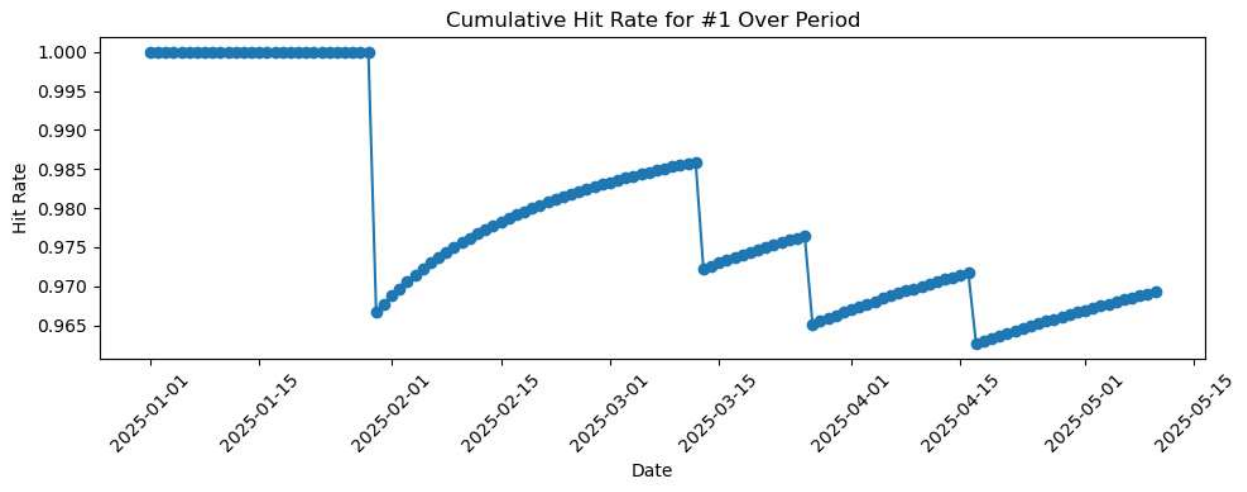

Figure 7: Cumulative Hit Rate for #1 Song Over Period
The plot shows periods of success interspersed with misses, reflecting the challenging nature of daily #1 predictions. The overall hit rate can be derived from this cumulative data.

---

8. RESULTS AND DISCUSSION

8.1. Model Performance Summary

The LightGBM model, trained on historical chart data, Spotify API metadata, and engineered momentum features, demonstrated a strong ability to predict the #1 song on Spotify USA for the following day.

- **Overall Predictive Power:** An ROC AUC of 0.97 indicates high discrimination.
- **Daily Performance:** Mean precision and recall for identifying the #1 song were both 0.68 over a 130-day rolling window evaluation.
- **Calibration:** The model's predicted probabilities are reasonably well-calibrated, offering a good basis for decision-making.

8.2. Key Feature Insights

The feature correlation analysis (Figure 1) revealed that:

- streams and rank are the most dominant predictors, as intuitively expected.
- stream_momentum and general Spotify popularity also provide valuable signals.
  This confirms that recent performance and overall track buzz are critical factors.

8.3. Implications for Kalshi Market Trading

The model's ability to generate calibrated probabilities *before* the Kalshi market settles presents a potential trading edge. If the model's predicted probability for a song significantly deviates from the market-implied probability (derived from Kalshi contract prices), a trader could identify overvalued or undervalued contracts.
For example, if the model predicts a 70% chance for Song A to be #1, but Kalshi's market prices imply only a 50% chance (i.e., "Yes" contract for Song A is at 50¢), a trader might buy "Yes" contracts. Conversely, if the market implies a 70% chance but the model predicts 50%, one might sell "Yes" contracts (or buy "No").
The daily Excel output generated by the notebook, listing songs with their predicted probabilities and actual next-day #1 status, serves as a direct input for developing and backtesting such trading strategies.

However, it's crucial to note, as per user point 1, that simply predicting if a song is #1 does not automatically translate into a profitable trading strategy. The actual profitability would depend on the Kalshi market's liquidity, the bid-ask spread, and how often the model's edge overcomes these transaction costs. A rigorous backtest against historical Kalshi market prices is necessary to quantify this.

---

9. CONCLUSIONS AND RECOMMENDATIONS

9.1. Conclusion

This project successfully developed an analytical tool utilizing API data, feature engineering, and a LightGBM classification model to predict the #1 song on Spotify USA. The model demonstrated strong predictive performance with an ROC AUC of 0.97 and reasonable daily precision/recall, indicating its potential to provide valuable insights for participants in the corresponding Kalshi prediction market. The core techniques employed, including API usage and predictive modeling, were effectively implemented.

9.2. Recommendations for the Analytical Tool

- **Develop a Backtesting Framework:** To assess true profitability, integrate historical Kalshi market data (prices and odds) to simulate trading based on model predictions and evaluate against actual market conditions (user point 1).
- **Expand Data Sources (Web Scraping & Other APIs):**
    - Incorporate social media sentiment (e.g., Twitter API, Reddit API via web scraping) for artists and tracks.

    - Scrape news mentions or music blog coverage.
    - Re-evaluate Google Trends integration with strategies to manage rate limits (e.g., less frequent updates, focusing on top contenders).
    - These additional data sources, leveraging web scraping and other techniques learned in class, can significantly bolster the pipeline and potentially improve model accuracy (user point 2).
- **Refine Feature Engineering:** Explore more complex features, such as interaction terms, longer-term trends, or features capturing artist-specific effects (e.g., an artist's historical "stickiness" at #1).
- **Model Calibration & Interpretation:** Further refine model calibration to ensure probabilities are highly reliable. Explore SHAP values or similar methods for better interpretability of daily predictions.
- **Monetization Strategy:** The developed data scraping/collection pipeline and predictive model can form the basis of a monetizable service by:
    - Offering subscription access to the analytical tool and its daily predictions.
    - Providing curated data feeds or API access to more sophisticated traders (point 2).

9.3. Limitations

- **Data Availability:** The model relies on the timeliness and availability of Spotify chart data and API data.
- **Google Trends Unfeasibility:** The inability to integrate Google Trends data due to rate limiting was a constraint (user point 3).
- **Static Feature Set:** The current feature set, while effective, does not yet incorporate broader sentiment or real-time news, which could impact sudden shifts in popularity.
- **No Live Trading/Backtesting:** The project focuses on prediction, not on live trading execution or a full financial backtest against Kalshi odds.

9.4. Future Work

- Implement a full financial backtesting module against historical Kalshi market data.

- Integrate additional data sources as recommended (social media, news, Genius API).
- Explore more advanced modeling techniques (e.g., ensemble methods, neural networks for richer feature interactions).
- Develop a user-friendly interface or dashboard for the analytical tool.
- Automate the entire data ingestion, feature engineering, prediction, and reporting pipeline for daily operation.

---

10. BIBLIOGRAPHY/REFERENCES

- Kalshi Prediction Markets: https://kalshi.com/
- Spotify Web API Documentation: https://developer.spotify.com/documentation/web-api/
- Spotipy (Python library for Spotify Web API): https://spotipy.readthedocs.io/
- Spotify Charts: https://charts.spotify.com/charts/overview/us
- LightGBM Documentation: https://lightgbm.readthedocs.io/
- Pandas Documentation: https://pandas.pydata.org/pandas-docs/stable/
- Scikit-learn Documentation: https://scikit-learn.org/stable/