Team 3 - 1.4; Topic 1: Enhancing the Song Recommender Engine from Module 11.4.

Ally Dong, Hannah Moreau, Varun Sivakumar, Sahieshnu Vasanthan.

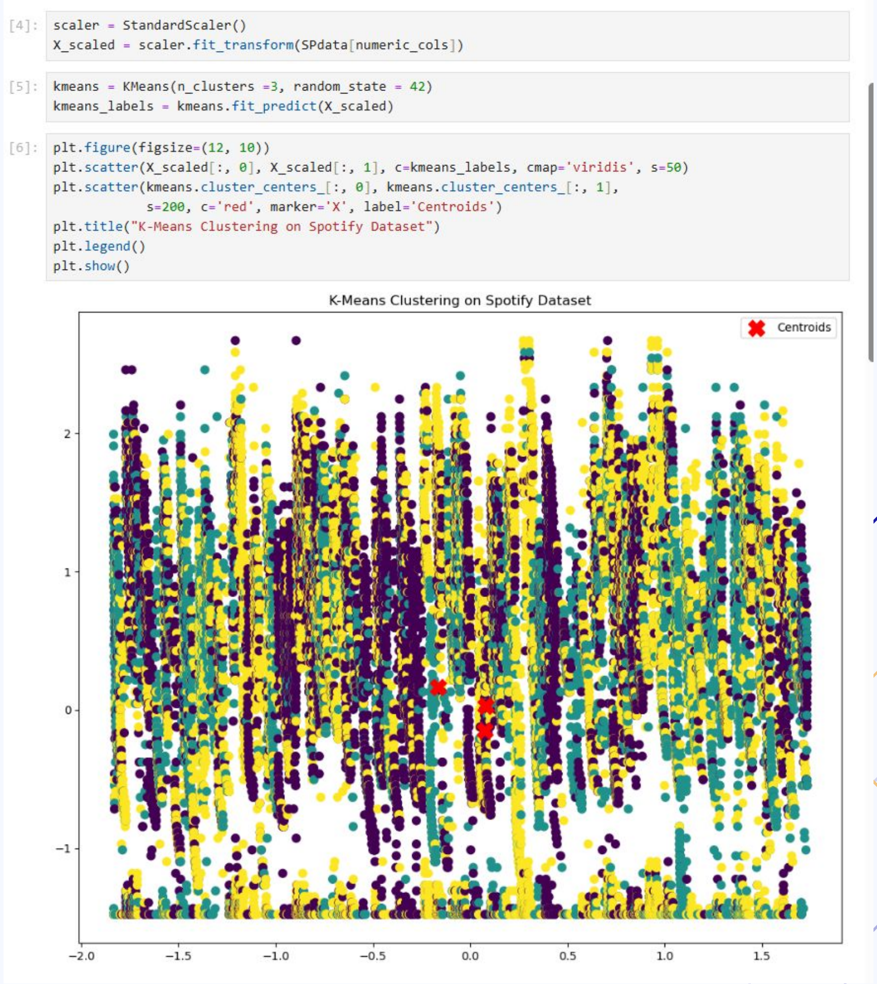
Table of contents

01

The Model & Enhancements.

The Model.

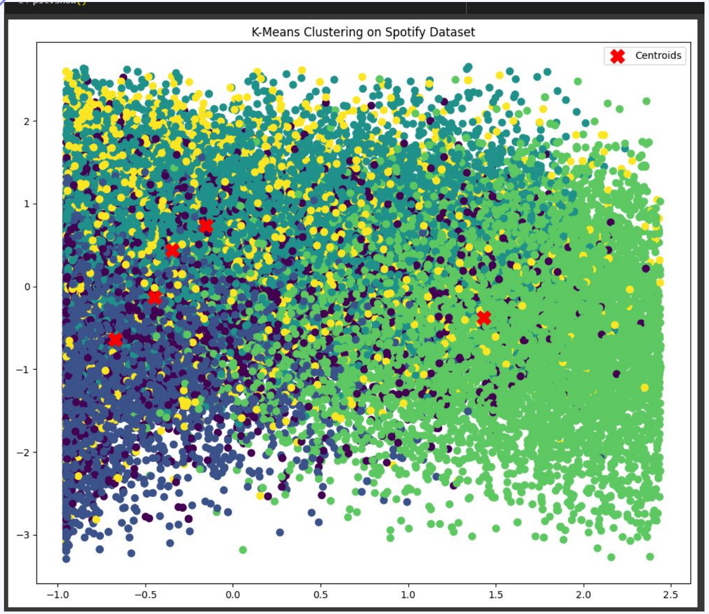
The model on the right shows the original K-Means model, and the plot.

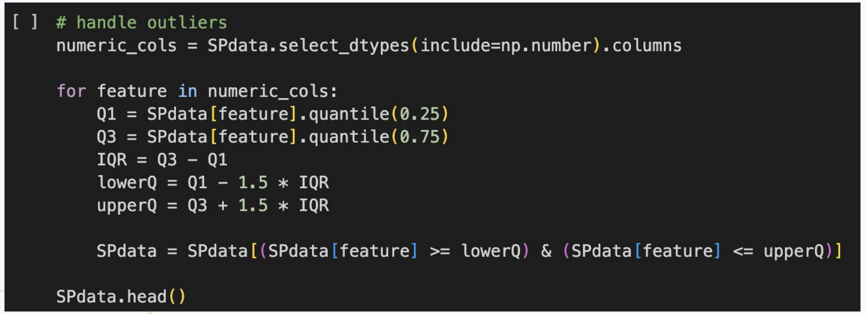


```
[4]:  scaler = StandardScaler()
      X_scaled = scaler.fit_transform(SPdata[numeric_cols])

[5]:  kmeans = KMeans(n_clusters =3, random_state = 42)
      kmeans_labels = kmeans.fit_predict(X_scaled)

[6]:  plt.figure(figsize=(12, 10))
      plt.scatter(X_scaled[:, 0], X_scaled[:, 1], c=kmeans_labels, cmap='viridis', s=50)
      plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
                  s=200, c='red', marker='X', label='Centroids')
      plt.title("K-Means Clustering on Spotify Dataset")
      plt.legend()
      plt.show()
```





```
[ ] # handle outliers
    numeric_cols = SPdata.select_dtypes(include=np.number).columns

    for feature in numeric_cols:
        Q1 = SPdata[feature].quantile(0.25)
        Q3 = SPdata[feature].quantile(0.75)
        IQR = Q3 - Q1
        lowerQ = Q1 - 1.5 * IQR
        upperQ = Q3 + 1.5 * IQR

        SPdata = SPdata[(SPdata[feature] >= lowerQ) & (SPdata[feature] <= upperQ)]

    SPdata.head()
```

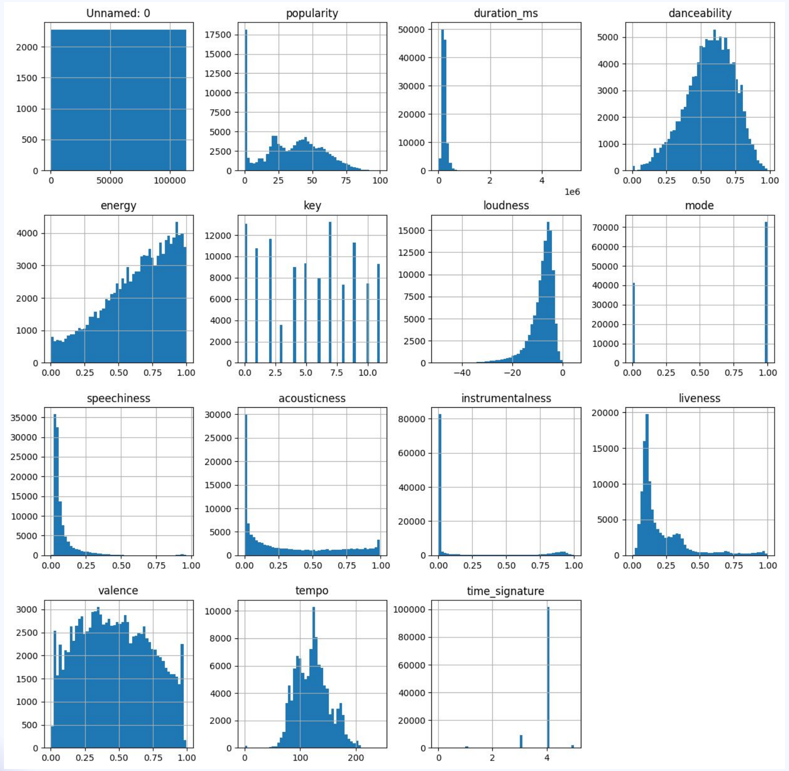
Enhancements & their Benefits

This is the enhanced model.

02

Data Exploration.

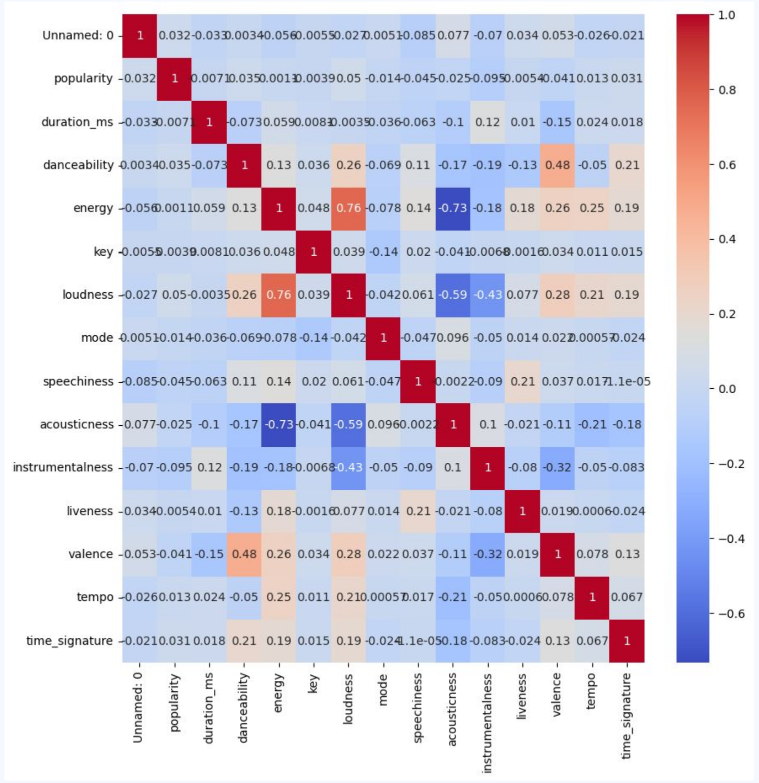
Histograms



Takeaways

- Most songs had a danceability of upper middle range
-Most songs were higher in energy
-Most songs had a medium tempo

Heat Map and Correlation



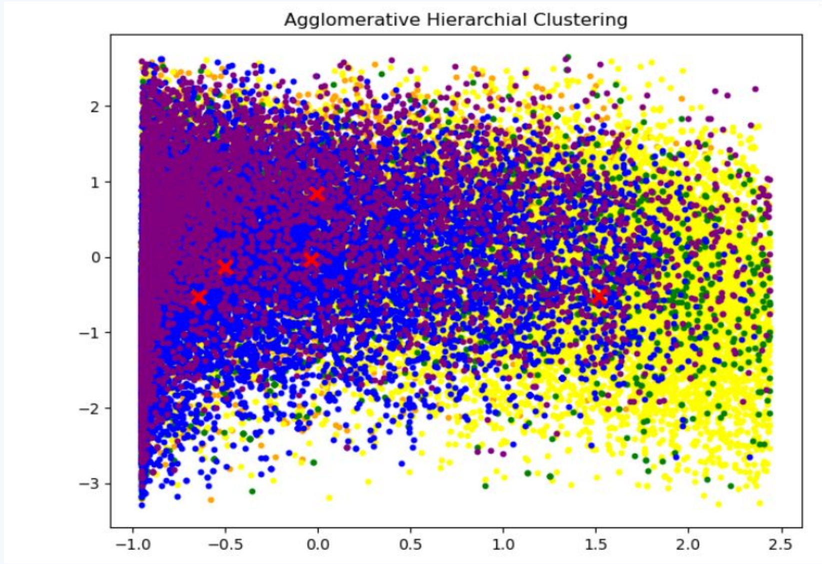
Takeaways

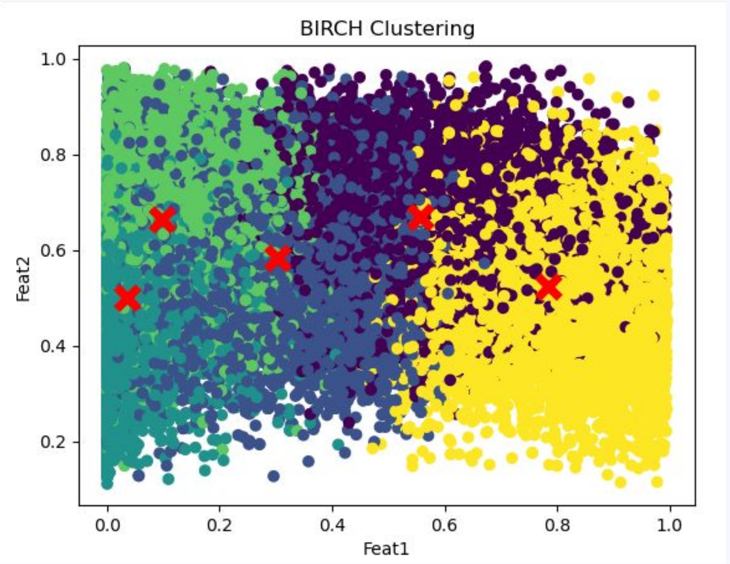
Most variables are not correlated or very lowly correlated. Loudness and energy are positively correlated, and energy and acoustic ness are negatively correlated. Using the genre of a song could also provide information.

03

Clustering Algorithms.

Compare & Contrast.





Agglomerative Hierarchical:
The agglomerative hierarchical clustering algorithm shows five clusters that are stacked on top of one another, and centroids indicating the average distance to the midpoint from all data points. There is clearly a noticeable amount of overlap which tells us that the data points belong to multiple clusters and can direct us towards more flexible clustering algorithms to get better results. This plot derives from considering each data point as a cluster and continuously merging into nearby clusters until the optimal cluster amount is reached.

BIRCH:
The BIRCH clustering algorithm shows five clusters that overlap surrounding clusters, and centroids indicating the center of the subcluster of data points. This tells us that there is distinction between the data points (the songs) yet some similarity due to the overlap where the data points fit between more than one cluster. This plot is created from the CF-tree (Clustering Feature tree) that organizes the data points which is part of the BIRCH process.

Advantages & Trait Comparison.

K Means	Agglomerative	DBSCAN	BIRCH
• Centroid-based • Within-cluster sum of squares • Sensitive to outliers	• Merging process • Requires linkage criteria • Computationally expensive • Allows use of distance metrics	• Auto-determines number of clusters • Non-linear complex clustering • Flexible towards outliers & noise	• Can handle large datasets • Uses CF-tree for memory efficiency • Flexible towards outliers & noise

All of the above work with unlabeled data.

04

User Feedback and Future Enhancements

How user feedback can be incorporated into the model in future and its impact

- Collect user ratings after song recommendations (e.g., Like / Dislike or 1–5 stars).
- Store feedback in a log with song features and cluster info.
- Use feedback to:
  Refine clustering over time (weighted relevance).
  Personalize future suggestions.
- Could explore reinforcement learning to adapt to user preferences dynamically.
- Result: A more accurate, personal, and responsive recommendation engine.

Future Enhancements that can be made to the model

📍 Genre-Based Filtering: Add genre awareness for more diverse or refined clusters.

🌍 User Profile Integration: Let users create profiles to track preferences.

🤖 Hybrid Models: Combine content-based filtering with collaborative filtering (like Spotify’s real engine).

📊 More Audio Features: Use additional Spotify features like key, mode, time signature for deeper clustering.

🎯 Real-Time Recommendation Tuning based on current mood/activity.

EXIT TICKET.

If you had to pick one, which future enhancement do you think would benefit users the best?

Thanks!

Any questions?

Team 3:
ally_dong@brown.edu
hannah_moreau@brown.edu
varun_sivakumar@brown.edu
sahieshnu_vasanthan@brown.edu

slidesgo