SAAVN Analytics Project

Submitted By Venkatesh Jagannthan

Objective

1) Classify song listeners based on their listening behaviors from the click stream data
2) Form clusters of users with liking for same artist
3) Applying notification data to cluster of users liking artist for that notifications
4) Compare against actual user click data to measure user click through rate

Approach

Step 1 Data Load

To achieve this, provided datasets are loaded

1. Sample Clickstream Data
2. Metadata to know artist id
3. NotificationClicks
4. Notification Artists

Step 2 Data Cleaning

Click Stream Data

1. Date column if formatted as yymmdd
2. timestamp column is dropped

Step 3 Data Processing

1. Clickstream data is combined with metadata to obtain artists the users listened to
2. Columns UserID, ArtistID, Date are indexed & vector assembled in pipeline mode
3. Bisecting K-Means Classification is used as preferred algorithm for this project. It requires to identify the initial K value by comparing WSSSE (Within Set Sum of Square Error) and choosing a point where value converges.

Command 1

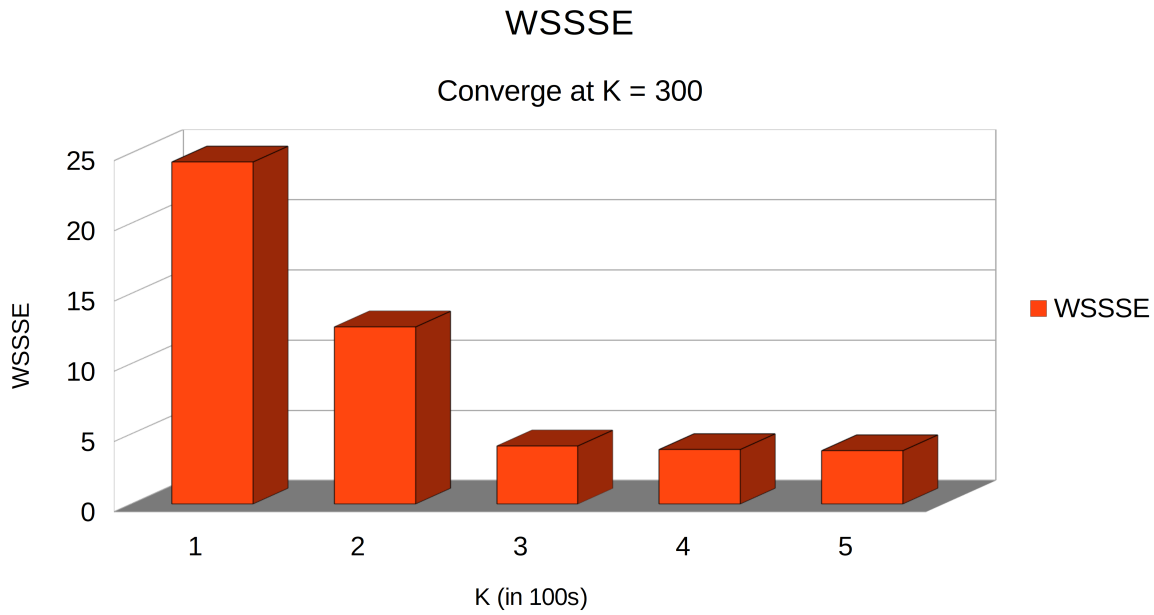
java -jar spark-submit notify.jar -findk
or
spark-submit –class NotificationDriver notify.jar -findk

*set AWS_ACCESS_KEY_ID & AWS_SECRET_ACCESS_KEY environment variable before running

Columns used for classification–
UserID,
ArtistID (of the song streamed)
Date (of song streamed)

WSSSE Values Chart: -

K	WSSSE
100	24.34
200	12.63
300	4.15
400	3.87
500	3.79



Choosen K Value = 300

4. Now programs is again run with chosen K value

Command 2

```
java -jar notify.jar -kmeans 300
or
spark-submit –class NotificationDriver notify.jar -kmeans 300
```

*set AWS_ACCESS_KEY_ID & AWS_SECRET_ACCESS_KEY environment variable before running

5. The bisecting K means provides predictions for user cohorts in the dataset **predictions.**
6. Cohorts with similar artist likings are grouped in dataset **cohorts**
7. Then for each cohort, artist with maximum listening count is chosen as the preferred artist of the cohort and stored in **preferredartists.**

8. The notification for each artist & users who actually clicked are obtained in **notifications** dataset
9. Then for all cohorts of users with same top artists a cluster is imagined & notification for that top artist applied to those all users. Results are obtained in dataset notifylist.
10. For each group of user for given artist, count of users to be notified are measured against actual clicks count (with help of left outer join) to get click through rate.
11. The expected output: -
    1. Top 5 Notifications with highest CTR – **CTR.csv**
    2. Notified Users & Artists for top 5 notifications – **Notifications/<notificationid>.csv**
    3. Intermediate result of all user cluster used to apply notification – **UserClusterArtist.csv**

Post Processing Steps

The above results are partitioned to a single output stream & files/folders renamed as per naming requirements.