Comparing K-Means and Alternating Least Squared Methods for the Purpose of Movie Recommendations

Ryan Jones – COP6526 – October 25, 2020

The scripts included, *kmeans.py* and *als.py*, can be run from the command line using the below commands. See the README file for more information on how to configure your system and run these files.

/spark/bin/spark-submit kmeans.py

/spark/bin/spark-submit als.py --driver-memory 4g als.py

Note that the data filepaths are defined at the top of the script, these can be changed manually to the correct directory on your machine. The output to screen is minimal. This is intended, as it reduces run-time and total processing. The user can change this and produce more print-to-screen by un-commenting the lines of code with *print()* statements.

K-Means Method and Results

K-Means clustering was performed for increasing values of K, number of clusters, from 2 to 20. The ideal K-value was determined to be 9. This number was chosen for it's comparatively good performance, as seen in the silhouette plot, and relatively low K.

Using $K = 9$, the algorithm was run and average ratings per user-cluster were calculated. These values were placed into a ratings matrix, along with actual ratings. Predicted user ratings for certain movies can be obtained from this matrix.

Evaluation of the method was performed and an RMSE of ~ 0.98 was calculated. No train-test split method, or cross validation method was used, so the resulting model may have overfitting characteristics.

Alternating Least Squares and Results

Alternating Least Squares (ALS) was performed for varying values of $rank$, $lambda$, and $numIters$. A train-test-validate split method was employed to reduce the effect of overfitting. The best model had values of $8, 0.1, 25$, for these parameters, respectively.

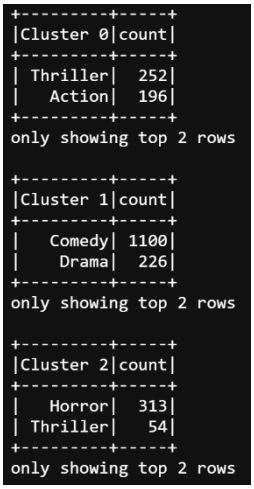
Using the best model parameters, the ALS method produced an RMSE of ~ 0.89, which is considerably lower than that produced by K-Means.

Comparison of K-Means and ALS

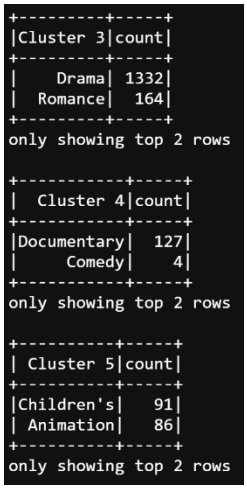
The K-Means model was created using the entire dataset, i.e. no train-test split or cross validation was used, so the results are likely prone to overfitting. Despite ALS being deployed with a train-test-validate split method, it still outperformed K-Means when using RMSE as an evaluation metric.

For the case of building a user recommendation system, ALS may be the better alternative because the underlying/hidden structure of the data is not necessary. Alternatively, K-Means produced "human

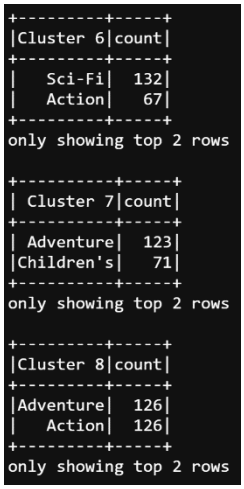
readable" results in that the clustering can be understood easily by a human. For example, clustering with $K = 9$ and identifying the two most frequently occurring movie categories for each cluster gives us an idea of the types of movies that are clustered together. This is more easily explained visually, as shown below.



```
+---------+-----+
|Cluster 0|count|
+---------+-----+
| Thriller|  252|
|   Action|  196|
+---------+-----+
only showing top 2 rows

+---------+-----+
|Cluster 1|count|
+---------+-----+
|   Comedy| 1100|
|    Drama|  226|
+---------+-----+
only showing top 2 rows

+---------+-----+
|Cluster 2|count|
+---------+-----+
|   Horror|  313|
| Thriller|   54|
+---------+-----+
only showing top 2 rows
```



```
+---------+-----+
|Cluster 3|count|
+---------+-----+
|    Drama| 1332|
|  Romance|  164|
+---------+-----+
only showing top 2 rows

+-----------+-----+
|  Cluster 4|count|
+-----------+-----+
|Documentary|  127|
|     Comedy|    4|
+-----------+-----+
only showing top 2 rows

+----------+-----+
| Cluster 5|count|
+----------+-----+
|Children's|   91|
| Animation|   86|
+----------+-----+
only showing top 2 rows
```



```
+---------+-----+
|Cluster 6|count|
+---------+-----+
|   Sci-Fi|  132|
|   Action|   67|
+---------+-----+
only showing top 2 rows

+----------+-----+
| Cluster 7|count|
+----------+-----+
| Adventure|  123|
|Children's|   71|
+----------+-----+
only showing top 2 rows

+---------+-----+
|Cluster 8|count|
+---------+-----+
|Adventure|  126|
|   Action|  126|
+---------+-----+
only showing top 2 rows
```

Cluster	***Description***
0	*Thriller*
1	*Comedy*
2	*Horror / Thriller*
3	*Drama / Romance*
4	*Documentary*
5	*Animated Children's*
6	*Sc-Fi / Action*
7	*Children's Adventure*
8	*Action / Adventure*

K-Means would be a better model to use in the case where such understanding is required for the specific use case.