WEEK 7 KMEANS

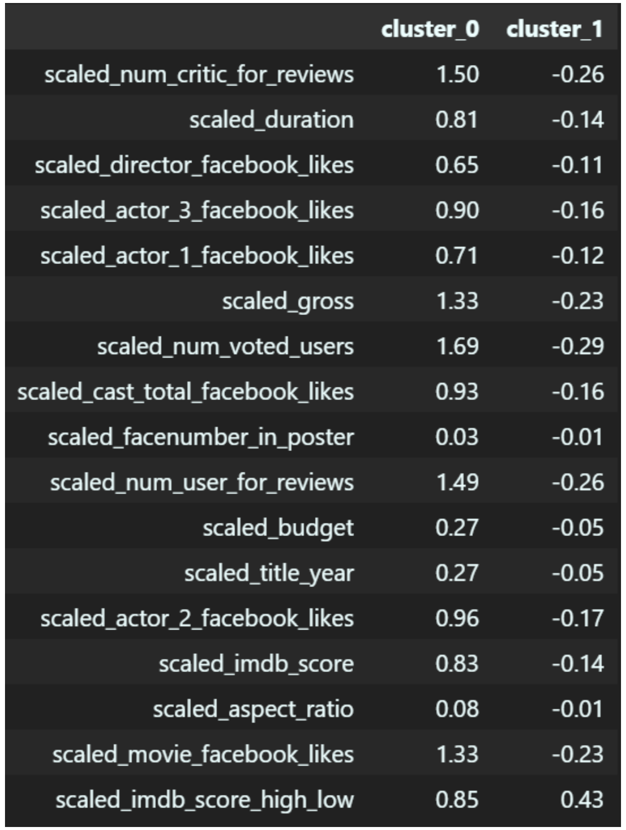
6438169421 Pattaradanai Lakkananithiphan

PREPROCESSING

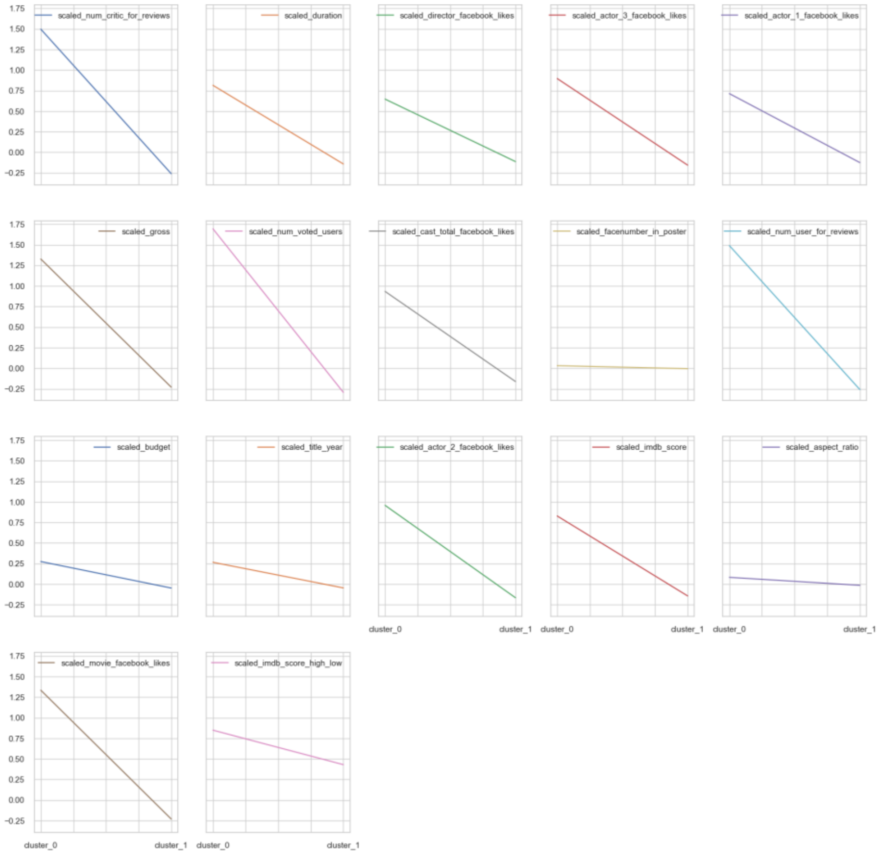
- Cut the middle 10% of the imdb_score column and use the median as separator to create a new column -> imdb_score_high_low
- Drop the categorical columns
- Encode int columns as float
- Fill in NaN with median in columns that makes sense to do so -> duration, gross, aspect_ratio / Drop those that are not
- Scale the data

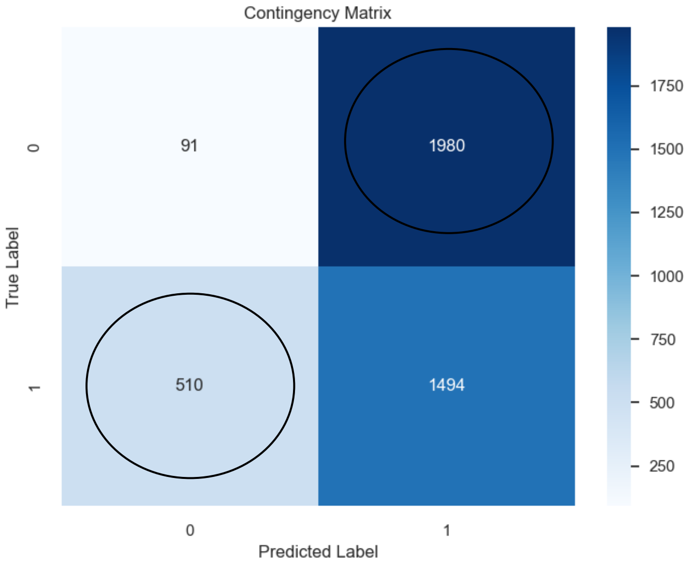
MODEL 1: K = 2

Cluster centers:



	cluster_0	cluster_1
scaled_num_critic_for_reviews	1.50	-0.26
scaled_duration	0.81	-0.14
scaled_director_facebook_likes	0.65	-0.11
scaled_actor_3_facebook_likes	0.90	-0.16
scaled_actor_1_facebook_likes	0.71	-0.12
scaled_gross	1.33	-0.23
scaled_num_voted_users	1.69	-0.29
scaled_cast_total_facebook_likes	0.93	-0.16
scaled_facenumber_in_poster	0.03	-0.01
scaled_num_user_for_reviews	1.49	-0.26
scaled_budget	0.27	-0.05
scaled_title_year	0.27	-0.05
scaled_actor_2_facebook_likes	0.96	-0.17
scaled_imdb_score	0.83	-0.14
scaled_aspect_ratio	0.08	-0.01
scaled_movie_facebook_likes	1.33	-0.23
scaled_imdb_score_high_low	0.85	0.43

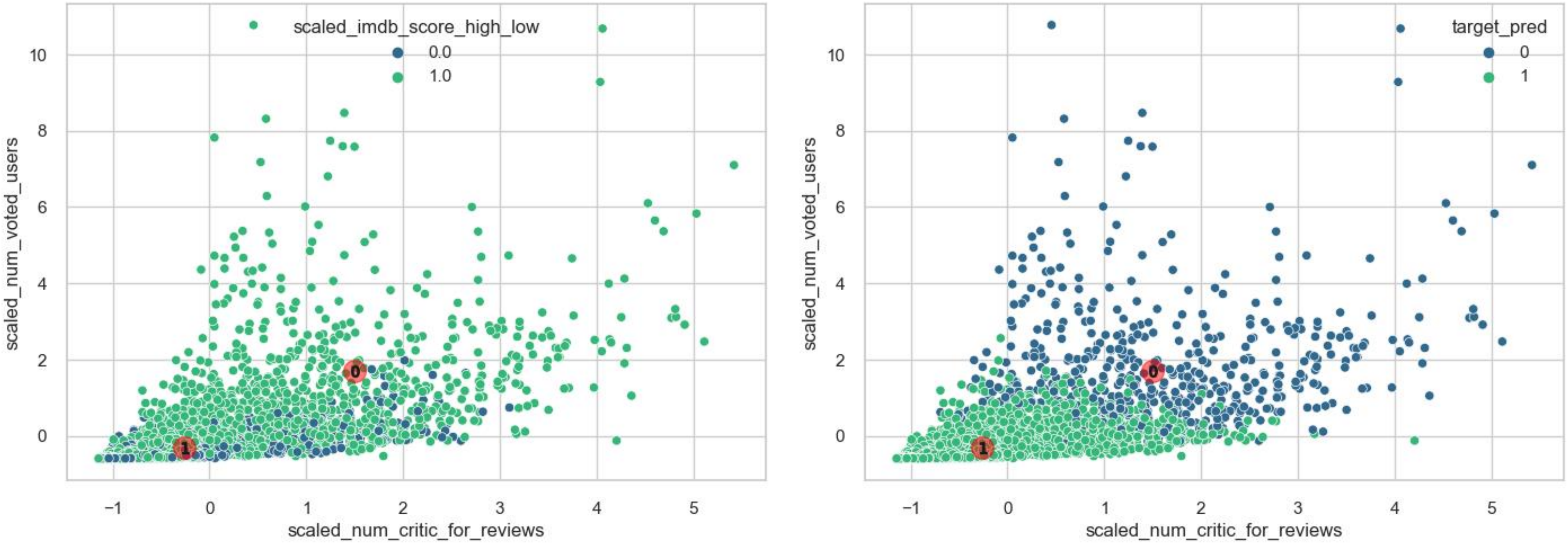




(the black circle is the 'correct' predictions)

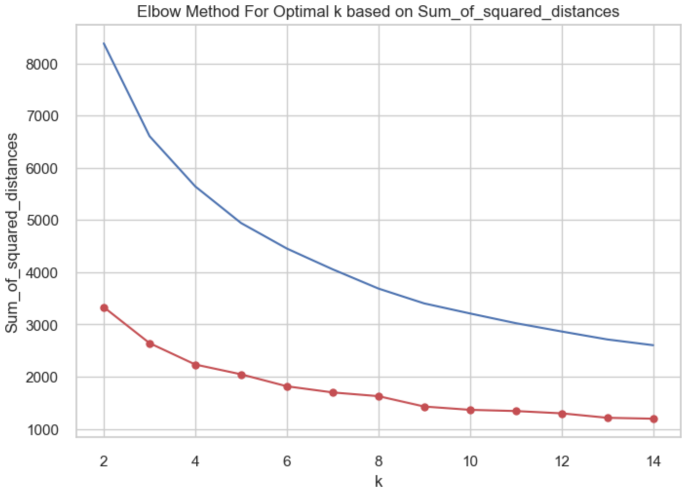
Score = -54919.28

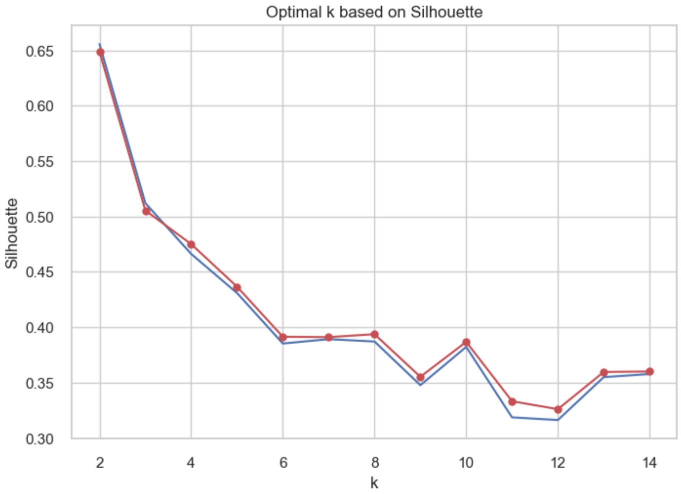
MODEL 1 (CONT.): SCATTER OF THE IMPORTANT FEATURES



Could be that the class flipped

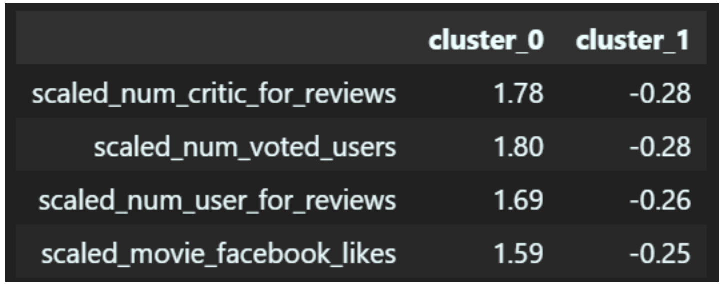
MODEL 2: WHAT K IS BEST => K = 2





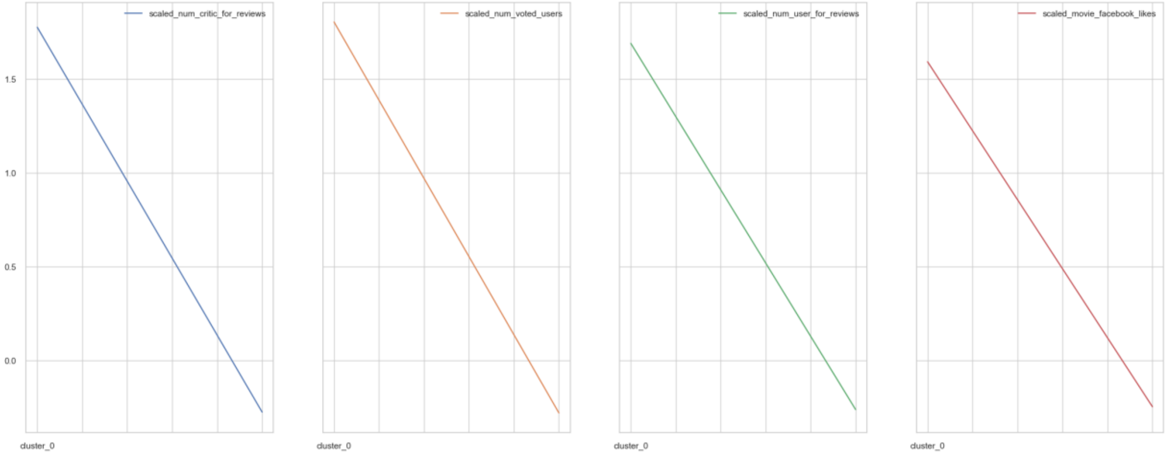
Since, K is the same, I tried selecting 4 most important features

Cluster centers:

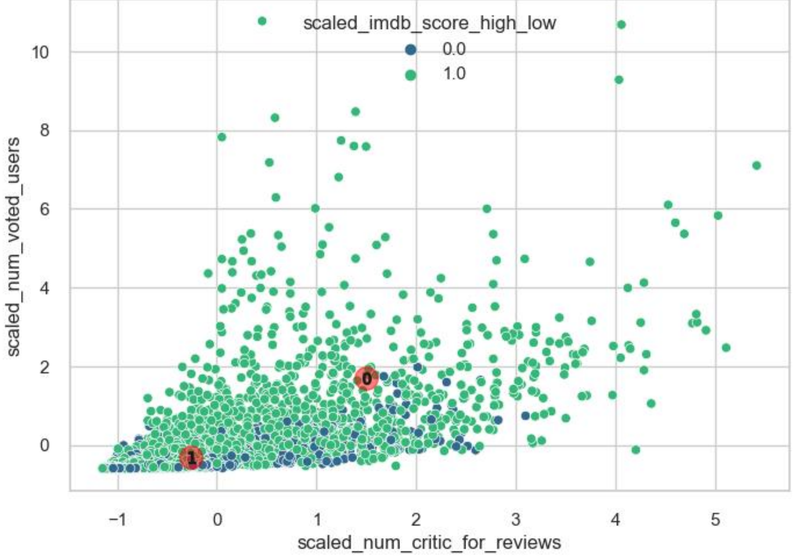


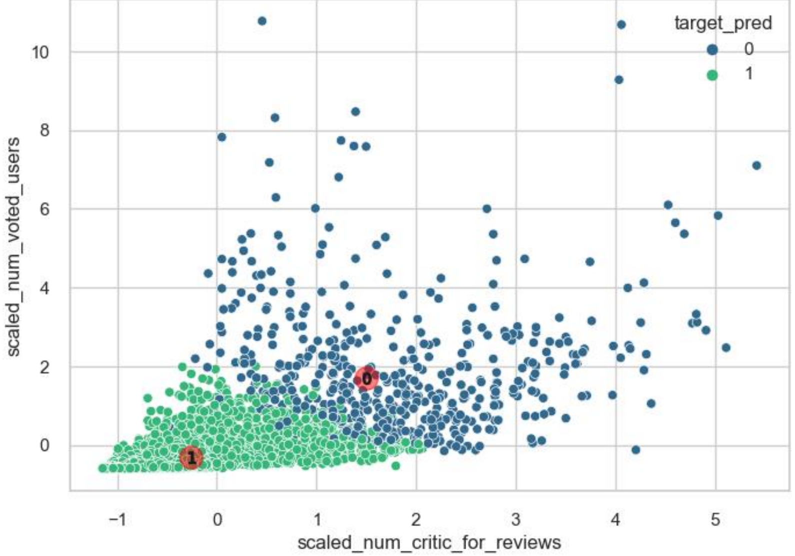
	cluster_0	cluster_1
scaled_num_critic_for_reviews	1.78	-0.28
scaled_num_voted_users	1.80	-0.28
scaled_num_user_for_reviews	1.69	-0.26
scaled_movie_facebook_likes	1.59	-0.25

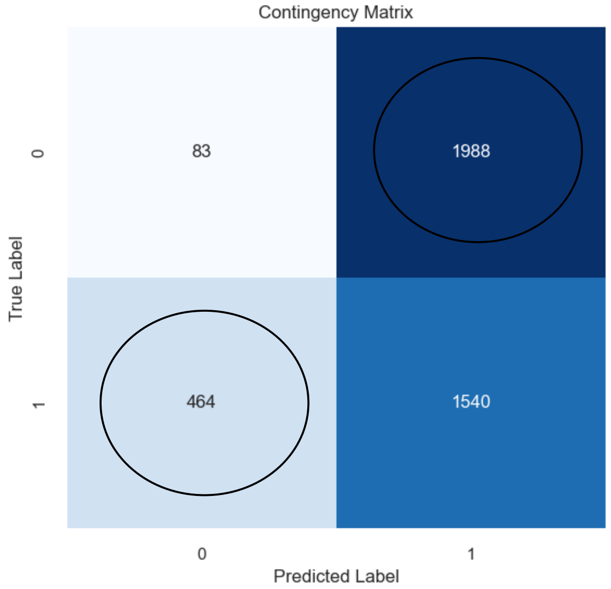
Score = -8841.6



MODEL 2 (CONT.): SCATTER OF THE IMPORTANT FEATURES







	0	1
0	83	1988
1	464	1540

The class number is flipped

EVALUATION

The performance of both models are approximately the same but the second one use much less features and have much better score, so we pick the

MODEL 2