Datasets:

1. Top 1000 channels by views
   Fields:
   YT_Title – channel title
   YT_id – channel youtube id
   YT_Username – channel username
   YT_category – channel category
   YT_Total_Subscribers – number of subscribers
   YT_Total_Views – number of views generated by a channel

2. 36 million events (likes, comments, subscriptions) generated by 17 million users on these top channels
   Fields:
   activity - event type (commented, subscribed, liked)
   author_id - user who did the action
   channel_id - channel that is related to the event (one of top 1000 channels)
   created_at - timestamp of the event

Description of the solution:

10 closest channels for each of 1000 channels go from three different sources:

1. **6** nearest neighbors based on data with activity 'subscribed':
   Nearest Neighbors implementation from scikit.
   Similarity Metrics – Jaccard.
   Users that are subscribed on more than 10 channels.

2. **2** nearest neighbors based on general information for 1000 channels:
   Features: 1699 topics, Country, Number of subscribers
   Similarity Metrics – Minkowski.

3. **2** nearest neighbors based on data with activity 'commented':
   Nearest Neighbors implementation from scikit.
   Similarity Metrics – Jaccard.
   Users that are commented on more than 30 channels.

Files:

Results1.docx –results of Model1, Results2.docx –results of Model2, Results3.docx –results of Model3.

Overview:

It is the task for similarity search.

With general information for channels, and information by audience, it makes most sense to use both datasets, so different information is included in final 10 nearest neighbors.

However, results of models 1 and 3 are based on audience preferences, while results of model 2 are based on general signals that we may want to add to our recommendations, e.g. we want to show channels from the same country, channels with the same topics. 3 separated models allow for different combinations. It is also possible to combine all the datasets with users and general information into one dataset and build one model on them, but it would be more difficult to interpret. Biggest weight was given to Model 1 which is based on data with action 'subscribed' as this type of action means the strongest connection between user and channel, and it can strongly reflect on similarities between channels, and lead to more accurate results (do users like recommendations or not).

Similarity metrics for models 1 and 3 is 'Jaccard similarity':

these models are built on only binary data (channels by users), so in this case it is very easy to interpret Jaccard distance between two vectors. With this similarity channels that have many subscribers and channels that have a few subscribers will be similar until they have many of them in common:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

While there are many other similarities it is very good one for binary data.

Similarity metrics for model 2 is Minkowski with p=2, which is the same as Euclidian. Besides the binary features (topics) we have 'scaled number of subscribers' and categorical feature 'country index' in the model 2. That's why it makes more sense to use more continuous metrics:

$$d^{MKD}(i,j) = \sqrt[\lambda]{\sum_{k=0}^{n-1} \left| y_{i,k} - y_{j,k} \right|^{\lambda}}$$

Size of the datasets:

Model1 – users subscribed on more than 10 channels (about 20K users),

Model2 – all 1000 channels,

Model3 – users commented on more than 30 channels (about 20K users).

Restrictions in Model1 and Model3 are due to complexity of calculations for bigger datasets. However, active users are representative set, and can be used for good predictions.

How to measure precision of results:

Try to create some logical hypothesis which is right for the closest neighbors, do not include it in the model directly, create accuracy measure, and test on this hypothesis. For instance:

- high percentage of the common set of topics for similar channels for Models without topics,

- similar number of subscribes or views for similar channels for Models without these ratios,

- high percentage of common users (or low jaccard distance) for similar channels for Models based on general information, and not on audience.

These hypotheses were the main source for testing accuracy of Models 1-3.

Moreover, to test precision of the Models we can try to get feedback from users, e.g. text comments, and use them as a test features. For instance, with our Model do we show channels with positive comments or with negative comments if our goal to recommend more positive channels.

Do A/B testing to test different models. Give different recommendations for separated clusters of users, and measure an increase in the number of views.

How models can be improved:

- Play with parameters and weights for different models,
- Play with features (which features are representative, how to scale, what to add),
- Use *locality sensitivity hashing* for the entire matrix channel-user, e.g. use LSHForest from scikit-learn which allows for finding approximate nearest neighbors with LSH built-in.
  Or divide all the channels on groups (by 10-100), and perform LSH technics for each group to decrease dimensionality of the channel-user matrix for each group, then combine reduced matrixes. This way we can count for all users, even those who are subscribed on one channel, and find more deeper connections between channels.
- Use Spark for investigating and working with huge matrixes. Also, LSH is implemented in Spark.
- Test a few models with A/B testing for different clusters of users. And see which model or which coefficients are better.