Notes DM 2

Questions:

* If you know certain features of the data in the test set, can you use that, or should you account for any possible potential test set? (user id’s)

Ideas:

* One mean score for click or booking bool over all people?
* Dates splitten into month and year
  + Month for seasons
  + Years for trends; dates of test set are later than of training set
* User id’s in test set are subset of user id’s in training set?
  + Or are they not distinguishable anymore?
  + Use this? Makes it possibly more accurate for this test set, but not for new users?
* test\_ids = set(test.user\_id.unique())
* train\_ids = set(train.user\_id.unique())
* intersection\_count = len(test\_ids & train\_ids)
* intersection\_count == len(test\_ids)
  + From the rank 15. Should return True
* Eventual score should be based on booked=5, clicked=1 and nothing=0? Or only 5 for booking? Then mapk with k=5 is possible.
* Nothing will correlate linearly with the target because it is based on an arbitrary choice of property id number.
* How many property id’s zijn er?
* Try a random forest? Probably not the best…
  + Try 100 binary classifiers (yes/no in this cluster) ipv 1 for 100 clusters?
* Try to score a type or cluster of properties for a certain destination first, then within those choose the best? 🡪 Only possible if “no one” books a hotel outside of the destination they searched for. (could improve speed of the algorithm as part of the dataset can be dropped immediately) 🡪 check with unique country id’s per search destination id 🡪 if not 1 country but a couple, we can still do all of those, it will still be less
* LambdaMART: is learn to rank, listwise approach

What? (Slides college)

* Information filtering system to predict user rating or preference of/for something
* Filtering information by determining relevance
* User based, item based, knowledge based(?) or user-item interaction based
* Collaborative filtering
  + Wisdom of the crowd
  + Based on preferences in the past 🡪 will be similar in the future
  + Output; rating of 1 item or top-N list of items
  + Find users like Alice who have rated the item and use average
    - By means of K-nearest neighbours:
      * Measure similarity between users with pearson correlation
      * Trial and error to find K
      * How to aggregate k rating values while avoiding a runtime that is too long?
  + Find items that are similar to the item and use Alice’s ratings for those
    - Use singular value decomposition (U and Vt enzo)
  + Problems:
    - How to rate new items and how to help new users? (“cold start problem”)
* Content-based recommendations
  + Use more information about the items
  + Learn user preferences
  + locate/recommend items that are "similar" to the user preferences
    - DICE coefficient
    - Use text (Piek’s lecture)

General:

* Look at distributions of variables
* Use different methods and their outcome as new columns and combine recommendations of methods.
* Use part of training set as validation set
* Use only part of training set to actually train?
* User based vs Item based
* Take running time into account
* Take into account that the file order is not random! When scores are equal, do not use the file order but random, in order to prevent overfitting.

Susan Li @ towardsdatascience.com:

* Nothing seems to correlate linearly 🡪 which models are not suitable?
* Combinations of search destinations, hotel country, hotel market will definitely help finding the hotel cluster
* Only booking is of interest?
* Naïve Bayes, K-nearest neighbours classifier, random forest classifier, Multi-class logistic regression, SVM classifier (time consuming but good performance)

Ashkan Ebadi thesis:

* users clustering, matrix factorization module, and hybrid recommender system

Owen Zhang from previous competition:

* Ensemble of Gradient boosting machines
* Kijk in de slides
  + <https://www.dropbox.com/sh/5kedakjizgrog0y/_LE_DFCA7J/ICDM_2013?preview=3_owen.pdf>

Example/walk through:

<https://www.dataquest.io/blog/kaggle-tutorial/>

Files on collaborative filtering uit college zijn pdf’s in de map

Logboek:

* Vrijdag 3 mei literature search for tactics, packages etc
* Maandag 6 mei idem
* Woensdag 8 mei idem