# WNS Analytics Wizard 2019

## Second place solution.

1. **A brief on the approach:**
    a. Using gradient boosting as machine learning model
    b. Stratified cross validation.
    c. User standard for click prediction problems features: (value counts and mean encoding by ids, different between click times, group by id, and length of unique items).
    d. Feature selections and fine tuning parameters of model.
2. **Data-preprocessing / feature engineering:**
    a. Standart features from train.csv:
        i. 'os_version',
        ii. 'is_4G',
        iii. 'value_counts_app_code',
        iv. 'mean_target_user_id' (mean value of target for every user_id in past time).
    b. User_id time features:
        i. Time from last and from next impression by user_id,
        ii. Minimum time between impressions by user_Ids
    c. User_id numbers features:
        i. Number of unique app_code for every user_id,
        ii. Difference between number of unique app_code and value_counts for evey user_id,
        iii. Number of unique items frow view_log.csv which the user was looking for time over a week ago(impression_time – 7 days),
        iv. number of user impressions in past and value_counts by user_ids in view_log.
    d. Group by app_code: For every app_code calculate men value by this features:
        i. 'mean_target_user_id'
        ii. 'value_counts_user_id',
        iii. time from from next impression by user_id,

    Features from a)-c) blocks its standard features for every is_click contest. They are always told about this feature in brief contests like this. A D-block feature was made because of high feature importance of 'value_counts_app_code' feature. So, I decide that need calculate some statistics by app_code. I am also don`t make time and number features by app code, I think it's can give me better score.

3. **My final model** it’s a mean rank (sort predictions from one model and give them rank from 0 to length test dataset) of prediction of 5 models, that’s make’s on different train set by stratified validation. Change validation from time series on stratified give me about 0.01 app in score. Also I drop features like app_code and user_id, although they gave the best score on validation. Some features, which calculate from view_log csv with using feature values, give a good score on validation. I also drop them. So, I have about 50 features. Then I calculate feature importance of gradient boosting for full set of features, sorted by them from high to low, and start drop features by one. In the end I get final subset of features described above.
4. **Takeaways**:
    a. Using stratified cross validation if distribution of target does not change in time.
    b. Using time and number statistics by ids.
    c. Using group by main ids.
    d. Drop feathers if you think they can overfit your model.
5. **Things a participant must focus on while solving such problems.**
    a. Choose the right validation.
    b. Start with a small set of features.
    c. Start Using Gradient Boosting.
    d. Make future selection.
    e. Check every step, if it allows the number of submissions.