

# Assignment 2

## Task 1

I used the CatBoost-Classifier, which can be called like this:

```
model = CatBoostClassifier(iterations=iterations, # Number of boosting iterations

                           learning_rate=learning_rate, # Learning rate
                           depth=depth, # Depth of trees
                           l2_leaf_reg=l2_leaf_reg, # L2 regularization strength
                           loss_function='MultiClass') # List of categorical features' indices
```

In order to get some good results, I ran some experiments on the training/test set with the following parameters:

```
with_or_without_scaled_features = [True, False]
feature_sets = [
    ["xDir", "yDir", "fixDensPerBB"],
    ["meanFix", "maxFix", "varFix"],
    ["meanFix", "maxFix", "varFix", "xDir", "yDir"],
    ["meanFix", "maxFix", "varFix", "xDir", "yDir",
     "fixDensPerBB"],
    ['blinkMean', 'blinkMin', 'blinkMax', 'blinkRate', 'xDir',
     'yDir', 'fixDensPerBB', 'duration', 'participant_id'],
    ['varFix', 'stdFix', 'meanDis', 'varDis', 'stdDisp',
     'freqDisPerSec',
     'number_of_blinks', 'blinkMean', 'blinkRate', 'xDir',
     'yDir', 'fixDensPerBB'],
    ['minFix', 'maxFix', 'minDis', 'maxDis', 'blinkMin',
     'blinkMax', 'xDir', 'yDir', 'fixDensPerBB'],
    ['minFix', 'maxFix', 'minDis', 'maxDis', 'blinkMin',
     'blinkMax', 'xDir', 'yDir', 'fixDensPerBB',
     'participant_id'],
    ['minFix', 'maxFix', 'varFix', 'stdFix', 'meanDis', 'minDis',
     'maxDis', 'varDis', 'stdDisp', 'freqDisPerSec', 'number_of_blinks',
     'blinkMean', 'blinkMin', 'blinkMax', 'blinkRate', 'xDir', 'yDir',
     'fixDensPerBB', 'duration', 'participant_id']]
iteration_list = [10, 100, 1000]
learning_rate_list = [0.01, 0.1, 0.5]
depth_list = [1, 2, 6]
l2_leaf_reg_list = [1, 3]

// Results:: Top 3 models
// key = str(using_scaling) + "_" + str(f_set) + "_" + str(iterations) + \
        "_" + str(learning_rate) + "_" + str(depth) + "_" + str(l2_leaf_reg)

("True_['minFix', 'maxFix', 'minDis', 'maxDis', 'blinkMin', 'blinkMax',
'xDir', 'yDir', 'fixDensPerBB']_100_0.1_2_1", 0.9090909090909091)
```

```
("True_['minFix', 'maxFix', 'minDis', 'maxDis', 'blinkMin', 'blinkMax',  
'xDir', 'yDir', 'fixDensPerBB']_100_0.1_2_3", 0.9090909090909091)  
("True_['minFix', 'maxFix', 'varFix', 'stdFix', 'meanDis', 'minDis', 'maxDis',  
'varDis', 'stdDisp', 'freqDisPerSec', 'number_of_blinks', 'blinkMean',  
'blinkMin', 'blinkMax', 'blinkRate', 'xDir', 'yDir', 'fixDensPerBB',  
'duration', 'participant_id']_10_0.01_6_3", 0.9090909090909091)
```

Interesting to see is, that also after only 10 iterations we can get good results. Also we can conclude that scaling helps a lot.

### Task 3 – context-aware assistant

My application is helping the user to focus on what she/he really wants to do. If he/she wants to read, then he/she can go to reading mode and it will help her/him with an alarm (text turning red), if she/he is not reading anymore. This should help to focus on a specific task. The same can be done for inspection & search.