Hyper parameter tuning scores

מקרא לפרמטרים מהמשתמש (מופיעים מעל התוצאות של כל מודל):

#RUN ARGUMENTS

args = {

"rs" : 42 ,#random state – קובע את ה'סיד' של הריצה כדי לשלוט ברנדומיות ולשמור על אותן תוצאות תחת אותם פרמטרים (בחרדנו רנדומית את 42, יכול היה להיות כל מספר)

"X\_version" :  1 ,# 1 = no dimension reduction (1000 features), 2 = medium dimension reduction (500 features), 3 = significant reduction (100-200 features)

"split\_rows" : 1 , # 1 = don't split rows, 2 = split to h1 and h7

"drop\_out\_correlated" : False ,

"age\_under\_50" : False ,

"debug" : False ,

"classification\_type": "normal", # "normal" =2 classes:  <50% , >50% change /"extreme" = 3 classes:  <30% , 30-70% , >70% change

"scoring\_method": "accuracy", # "accuracy" / "precision"/"recall'/"all\_scores".

"both": False, # True- train on both research 1 and research 2 data , False- train on research 2 only.IMPORTANT: if set to True, use X\_version = 1 only!!!

"cv":5, # param for cv

"n\_iter": 100 # param for cv

}

מקרא לפרמטרים של הקרוס ולידציה (ה'גריד') -דוגמה עבור פייפליין שבו הקלסיפייר (השלב הסופי)הוא רגרסיה לוגיסטית, והשלב המקדים הוא pca

param1a = { #LOGISTIC REGRESSION with pca, no selectkbest

    "pca\_\_n\_components": range(2,10), - -שם השלב בפייפליין שהפרמטר שייך לו, מימין- הפרמטר, מימין לנקודותיים, הערכים השונים שמנסים לשים בפרמטר הזה בכל איטרציה של הקרוס ולידציה משמאל ' \_\_ '

    "classifier\_\_C": np.logspace(-4, 4,50),

    "classifier\_\_penalty": ['l1','l2'],

    "classifier" : [clf1]

}

מקרא לתוצאות- דוגמה עבור התוצאה של רגרסיה לוגיסטית בתור השלב האחרון בפייפליין שלפניה יש כשלב מקדים בחירת פיצ'רים ע"י סלקט בסט קיי:

Fitting 4 folds for each of 8 candidates, totalling 32 fits

=== user arguments ===

 הפרמטרים מהמשתמש (מקרא למעלה)

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 4, 'n\_iter': 50}

=== Pipe ===

 תיאור הפייפליין- קודם כל תיקננו לערכים סטנדרטיים ע"י סקיילר, אחרכך הרצנו את קיי בסט, אחרכך השמטנו פיצ'רים עם קורלציה גבוהה אחד לשני (לא באמת עשינו את זה כי הפרמטר

Drop\_out\_correlated מהמשתמש

מוגדר להיות

(False

ואז הרצנו את הקלסיפייר- רגרסיה לוגיסטית

 Pipeline(steps=[('scaler', StandardScaler()), ('kBest', SelectKBest()),

                ('corr\_drop', CorrelationDropper2()),

                ('classifier',

                 LogisticRegression(C=0.046415888336127774, max\_iter=10000,

                                    random\_state=42))])

=== best hyperparametes picked in cv (cv's best score) ===

פה מתוארים לפנינו מבין 8 המועמדים לסטים של פרמטרים שהגריד סרץ' ניסה, מי היה הכי מוצלח:

 {'kBest\_\_score\_func': <function mutual\_info\_classif at 0x000001D097540280>, 'kBest\_\_k': 12, 'classifier\_\_C': 0.046415888336127774, 'classifier':

LogisticRegression(C=0.046415888336127774, max\_iter=10000, random\_state=42)}

פה מתוארים הפרמטרים שקיי בסט בחר בתוך הסט הכי טוב, בגלל שלא השתמשנו ב

Drop\_out\_correlated

אז הפרמטרים האלה הם בעצם הפרמטרים שעליהם המודל התאמן

אותם צריך להשוות לספרות ולדון במסקנות

=== feature selection proccess (cv's best score) ===

first feature selection (high corellation to y)

K best selected:

 ['Treatment\_group', 'Ratio\_Relative\_Power\_Norm\_Beta1 Over Beta2\_PO4', 'Ratio\_Relative\_Power\_Norm\_Beta1 Over Beta2\_T7', 'Ratio\_Relative\_Power\_Norm\_Delta Over Alpha\_PO3', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta2\_P4', 'Ratio\_Relative\_Power\_Norm\_Delta Over Theta\_F6', 'Ratio\_Relative\_Power\_Norm\_Theta Over Alpha\_FT8', 'Ratio\_Relative\_Power\_Norm\_Theta Over Alpha\_P5', 'Single\_Mean\_Power\_Abs\_Beta1\_Pz', 'Single\_Mean\_Power\_Abs\_Beta1\_T8', 'Single\_Mean\_Power\_Abs\_Delta\_FC1', 'Single\_Relative\_Power\_Norm\_Theta\_C2']

וזו התוצאה של הסט הכי טוב עם סטיית תקן. הציון הוא לפי מה שכתוב בשדה

Scoring\_method

בפרמטרים מהמשתמש

=== score (cv's best score): 0.828 +/- 0.048 ===

הרצות:

1. ניסיון 1: 24.12

Parameter grids to try in grid search:

param1a = { #LOGISTIC REGRESSION with pca, no selectkbest

    "pca\_\_n\_components": range(2,10),

    "classifier\_\_C": np.logspace(-4, 4,50),

    "classifier\_\_penalty": ['l1','l2'],

    "classifier" : [clf1]

}

param1b = {#LOGISTIC REGRESSION with selectkbest, no pca

    "classifier\_\_C": np.logspace(-4, 4, 4), #classifier (logistic regression) param 'C' for tuning

    "kBest\_\_k": [12], #selctKbest param 'k'for tuning. must be  <= num of features

    "kBest\_\_score\_func" : [mutual\_info\_classif,f\_classif], #selctKbest param 'score\_func'for tuning

    "classifier" : [clf1] # the classifier clf1 (LogisticRegression) will use as the final step in pipleine- the 'estimator'

}

param2 = { #KNN

    "classifier\_\_n\_neighbors" : [5,10,15],

    "kBest\_\_k": [i for i in range(4)],

    "kBest\_\_score\_func" : [mutual\_info\_classif,f\_classif],

    "classifier" : [clf2]

}

param3 = { #SVC

    'classifier\_\_gamma': [0.1,0.5,1,1.5,2],#old =  [10,6,3,1, 0.1, 0.01, 0.001, 0.0001]-> selcted = 1, score =0.6529411764705882

    'classifier\_\_kernel': ['rbf','sigmoid','linear','poly'],

    'classifier\_\_C':[0.1,1.10,100,1000],

    "kBest\_\_k": [2,5,8,12,16,20,30,45], #  k should be smaller than num of features always

    "kBest\_\_score\_func" : [mutual\_info\_classif,f\_classif],

    "classifier" : [clf3]

}

param4 = { # DECISION TREE

        'classifier\_\_max\_leaf\_nodes': list(range(2, 100)),

        'classifier\_\_min\_samples\_split': [2, 3, 4], #reason I tried this classifier params https://medium.com/analytics-vidhya/decisiontree-classifier-working-on-moons-dataset-using-gridsearchcv-to-find-best-hyperparameters-ede24a06b489

            "kBest\_\_k": [2,5,8,12,16,20,30,45],

            "kBest\_\_score\_func" : [mutual\_info\_classif,f\_classif],

            "classifier" : [clf4]

}

param5 = { # RANDOM FOREST

# reason I tried this classifier params:

# https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

        'classifier\_\_bootstrap': [True],

        'classifier\_\_max\_depth': [10, 50, 100, 200],

        'classifier\_\_max\_features': [2,5,10,20,50],

        'classifier\_\_min\_samples\_leaf': [2, 5,10],

        'classifier\_\_min\_samples\_split': [2,3,4],

        'classifier\_\_n\_estimators': [10,50,80,120,300,1000], #reason I tried this classifier params https://medium.com/analytics-vidhya/decisiontree-classifier-working-on-moons-dataset-using-gridsearchcv-to-find-best-hyperparameters-ede24a06b489

        "kBest\_\_k": [2,5,8,12,16,20,30,45],

        "kBest\_\_score\_func" : [mutual\_info\_classif,f\_classif],

        "classifier" : [clf5]

}

param6 = { #GRADIENT BOOSTING

# reason I tried this classifier params:

#https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/

    'classifier\_max\_depth':range(5,16,2),

    'classifier\_min\_samples\_split':range(200,1001,200),

    "classifier": [clf6]

}

param7 = { #CATBOOST CLASSIFIER

    'classifier\_\_depth': [3,7,10],

    'classifier\_\_learning\_rate': [0.1],

    "kBest\_\_k": [5,10,15],

    "classifier": [clf7]

}

Results for each one of the grids:

Fitting 4 folds for each of 50 candidates, totalling 200 fits

=== user arguments ===

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 4, 'n\_iter': 50}

=== Pipe === פייפ מספר 1

 Pipeline(steps=[('scaler', StandardScaler()), ('pca', PCA()),

                ('corr\_drop', CorrelationDropper()),

                ('classifier',

                 LogisticRegression(C=0.00014563484775012445, max\_iter=10000,

                                    random\_state=42))])

=== best hyperparametes picked in cv (cv's best score) ===

 {'pca\_\_n\_components': 8, 'classifier\_\_penalty': 'l2', 'classifier\_\_C': 0.00014563484775012445, 'classifier': LogisticRegression(C=0.00014563484775012445, max\_iter=10000, random\_state=42)}

=== feature selection proccess (cv's best score) ===

=== score (cv's best score): 0.621 +/- 0.019 === mציון של הפרמטרים הכי טובים שנבחרו מהגריד

Fitting 4 folds for each of 8 candidates, totalling 32 fits

=== user arguments ===

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 4, 'n\_iter': 50}

=== Pipe === פייפ מספר 2

 Pipeline(steps=[('scaler', StandardScaler()), ('kBest', SelectKBest()),

                ('corr\_drop', CorrelationDropper2()),

                ('classifier',

                 LogisticRegression(C=0.046415888336127774, max\_iter=10000,

                                    random\_state=42))])

=== best hyperparametes picked in cv (cv's best score) ===

 {'kBest\_\_score\_func': <function mutual\_info\_classif at 0x000001D097540280>, 'kBest\_\_k': 12, 'classifier\_\_C': 0.046415888336127774, 'classifier': LogisticRegression(C=0.046415888336127774, max\_iter=10000, random\_state=42)}

=== feature selection proccess (cv's best score) ===

first feature selection (high corellation to y)

K best selected: המשתנים שהסינון בסוף בחר להשאיר (קיי בסט עושה את הסינון). בגלל שהדרופ-אאוט-קורילייטד בארגומנטים מהיוזר הוא על פולס, הם המשתנים הסופיים שהקלסיפייר התאמן עליהם

 ['Treatment\_group', 'Ratio\_Relative\_Power\_Norm\_Beta1 Over Beta2\_PO4', 'Ratio\_Relative\_Power\_Norm\_Beta1 Over Beta2\_T7', 'Ratio\_Relative\_Power\_Norm\_Delta Over Alpha\_PO3', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta2\_P4', 'Ratio\_Relative\_Power\_Norm\_Delta Over Theta\_F6', 'Ratio\_Relative\_Power\_Norm\_Theta Over Alpha\_FT8', 'Ratio\_Relative\_Power\_Norm\_Theta Over Alpha\_P5', 'Single\_Mean\_Power\_Abs\_Beta1\_Pz', 'Single\_Mean\_Power\_Abs\_Beta1\_T8', 'Single\_Mean\_Power\_Abs\_Delta\_FC1', 'Single\_Relative\_Power\_Norm\_Theta\_C2']

=== score (cv's best score): 0.828 +/- 0.048 ===

=== user arguments ===

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 4, 'n\_iter': 50}

=== Pipe === פייפ מספר 3

 Pipeline(steps=[('scaler', StandardScaler()), ('kBest', SelectKBest()),

                ('corr\_drop', CorrelationDropper2()),

                ('classifier', KNeighborsClassifier(n\_neighbors=15))])

=== best hyperparametes picked in cv (cv's best score) ===

 {'kBest\_\_score\_func': <function mutual\_info\_classif at 0x000002067A440280>, 'kBest\_\_k': 2, 'classifier\_\_n\_neighbors': 15, 'classifier': KNeighborsClassifier(n\_neighbors=15)}

=== feature selection proccess (cv's best score) ===

first feature selection (high corellation to y)

K best selected:

 ['Treatment\_group', 'Ratio\_Relative\_Power\_Norm\_Beta1 Over Beta2\_T7']

=== score (cv's best score): 0.885 +/- 0.022 === ציון של הפרמטרים הכי טובים שנבחרו מהגריד

=== user arguments ===

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 4, 'n\_iter': 50}

=== Pipe ===

 Pipeline(steps=[('scaler', StandardScaler()), ('kBest', SelectKBest()),

                ('corr\_drop', CorrelationDropper2()),

                ('classifier',

                 SVC(C=0.1, gamma=1.5, kernel='poly', probability=True,

                     random\_state=42))])

=== best hyperparametes picked in cv (cv's best score) ===

 {'kBest\_\_score\_func': <function mutual\_info\_classif at 0x000002067A440280>, 'kBest\_\_k': 2, 'classifier\_\_kernel': 'poly', 'classifier\_\_gamma': 1.5, 'classifier\_\_C': 0.1, 'classifier': SVC(C=0.1, gamma=1.5, kernel='poly', probability=True, random\_state=42)}

=== feature selection proccess (cv's best score) ===

first feature selection (high corellation to y)

K best selected:

 ['Treatment\_group', 'Ratio\_Relative\_Power\_Norm\_Beta1 Over Beta2\_T7']

=== score (cv's best score): 0.874 +/- 0.037 === ציון של הפרמטרים הכי טובים שנבחרו מהגריד

=== user arguments ===

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 4, 'n\_iter': 50}

=== Pipe ===

 Pipeline(steps=[('scaler', StandardScaler()), ('kBest', SelectKBest()),

                ('corr\_drop', CorrelationDropper2()),

                ('classifier',

                 DecisionTreeClassifier(max\_leaf\_nodes=2, min\_samples\_split=4,

                                        random\_state=42))])

=== best hyperparametes picked in cv (cv's best score) ===

 {'kBest\_\_score\_func': <function f\_classif at 0x000002067A41D480>, 'kBest\_\_k': 2, 'classifier\_\_min\_samples\_split': 4, 'classifier\_\_max\_leaf\_nodes': 2, 'classifier': DecisionTreeClassifier(max\_leaf\_nodes=2, min\_samples\_split=4, random\_state=42)}

=== feature selection proccess (cv's best score) ===

first feature selection (high corellation to y)

K best selected:

 ['Treatment\_group', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta2\_Fp1']

=== score (cv's best score): 0.874 +/- 0.037 ===

=== user arguments ===

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 4, 'n\_iter': 50}

=== Pipe ===

 Pipeline(steps=[('scaler', StandardScaler()), ('kBest', SelectKBest()),

                ('corr\_drop', CorrelationDropper2()),

                ('classifier',

                 RandomForestClassifier(max\_depth=100, max\_features=20,

                                        min\_samples\_leaf=10,

                                        min\_samples\_split=3, n\_estimators=120,

                                        random\_state=42))])

=== best hyperparametes picked in cv (cv's best score) ===

 {'kBest\_\_score\_func': <function mutual\_info\_classif at 0x000002067A440280>, 'kBest\_\_k': 30, 'classifier\_\_n\_estimators': 120, 'classifier\_\_min\_samples\_split': 3, 'classifier\_\_min\_samples\_leaf': 10, 'classifier\_\_max\_features': 20, 'classifier\_\_max\_depth': 100, 'classifier\_\_bootstrap': True, 'classifier': RandomForestClassifier(max\_depth=100, max\_features=20, min\_samples\_leaf=10,

                       min\_samples\_split=3, n\_estimators=120, random\_state=42)}

=== feature selection proccess (cv's best score) ===

first feature selection (high corellation to y)

K best selected:

 ['Treatment\_group', 'age', 'Pair\_Mean\_Power\_Abs\_Delta\_O1-O2', 'Pair\_Mean\_Power\_Abs\_Delta\_T7-T8', 'Pair\_Mean\_Power\_Abs\_Theta\_F7-F8', 'Ratio\_Relative\_Power\_Norm\_Beta1 Over Beta2\_PO4', 'Ratio\_Relative\_Power\_Norm\_Beta1 Over Beta2\_T7', 'Ratio\_Relative\_Power\_Norm\_Delta Over Alpha\_PO3', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta1\_PO8', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta2\_P4', 'Ratio\_Relative\_Power\_Norm\_Delta Over Theta\_F6', 'Ratio\_Relative\_Power\_Norm\_Theta Over Alpha\_FT8', 'Ratio\_Relative\_Power\_Norm\_Theta Over Alpha\_P5', 'Ratio\_Relative\_Power\_Norm\_Theta Over Beta1\_C3', 'Single\_Mean\_Power\_Abs\_Beta1\_FC5', 'Single\_Mean\_Power\_Abs\_Beta1\_Pz', 'Single\_Mean\_Power\_Abs\_Beta1\_T8', 'Single\_Mean\_Power\_Abs\_Beta2\_O2', 'Single\_Mean\_Power\_Abs\_Delta\_FC1', 'Single\_Mean\_Power\_Abs\_Delta\_PO4', 'Single\_Mean\_Power\_Abs\_Delta\_T7', 'Single\_Relative\_Power\_Norm\_Beta1\_C2', 'Single\_Relative\_Power\_Norm\_Beta1\_C3', 'Single\_Relative\_Power\_Norm\_Beta1\_FT8', 'Single\_Relative\_Power\_Norm\_Beta2\_CP4', 'Single\_Relative\_Power\_Norm\_Beta2\_P6', 'Single\_Relative\_Power\_Norm\_Delta\_P5', 'Single\_Relative\_Power\_Norm\_Delta\_Pz', 'Single\_Relative\_Power\_Norm\_Theta\_C2', 'Single\_Relative\_Power\_Norm\_Theta\_CPz']

=== score (cv's best score): 0.874 +/- 0.037 ===

ניסיון 2- 24.12.2022

Fitting 5 folds for each of 12 candidates, totalling 60 fits

=== user arguments ===

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 5, 'n\_iter': 100}

=== Pipe ===

 Pipeline(steps=[('scaler', StandardScaler()), ('pca', PCA()),

                ('corr\_drop', CorrelationDropper2()),

                ('classifier',

                 LogisticRegression(C=0.0001, max\_iter=10000,

                                    random\_state=42))])

=== best hyperparametes picked in cv (cv's best score) ===

 {'pca\_\_n\_components': 7, 'classifier\_\_penalty': 'l2', 'classifier\_\_C': 0.0001, 'classifier': LogisticRegression(C=0.0001, max\_iter=10000, random\_state=42)}

=== feature selection proccess (cv's best score) ===

=== score (cv's best score): 0.621 +/- 0.023 ===

Fitting 5 folds for each of 32 candidates, totalling 160 fits

=== user arguments ===

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 5, 'n\_iter': 100}

=== Pipe ===

 Pipeline(steps=[('scaler', StandardScaler()), ('kBest', SelectKBest()),

                ('corr\_drop', CorrelationDropper2()),

                ('classifier',

                 LogisticRegression(C=21.54434690031882, max\_iter=10000,

                                    random\_state=42))])

=== best hyperparametes picked in cv (cv's best score) ===

 {'kBest\_\_score\_func': <function mutual\_info\_classif at 0x0000023729210280>, 'kBest\_\_k': 11, 'classifier\_\_C': 21.54434690031882, 'classifier': LogisticRegression(C=21.54434690031882, max\_iter=10000, random\_state=42)}

=== feature selection proccess (cv's best score) ===

first feature selection (high corellation to y)

K best selected:

 ['Treatment\_group', 'Ratio\_Relative\_Power\_Norm\_Beta1 Over Beta2\_PO4', 'Ratio\_Relative\_Power\_Norm\_Beta1 Over Beta2\_T7', 'Ratio\_Relative\_Power\_Norm\_Delta Over Alpha\_PO3', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta2\_P4', 'Ratio\_Relative\_Power\_Norm\_Delta Over Theta\_F6', 'Ratio\_Relative\_Power\_Norm\_Theta Over Alpha\_FT8', 'Ratio\_Relative\_Power\_Norm\_Theta Over Alpha\_P5', 'Single\_Mean\_Power\_Abs\_Beta1\_Pz', 'Single\_Mean\_Power\_Abs\_Beta1\_T8', 'Single\_Mean\_Power\_Abs\_Delta\_FC1']

=== score (cv's best score): 0.875 +/- 0.082 ===

Fitting 5 folds for each of 40 candidates, totalling 200 fits

=== user arguments ===

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 5, 'n\_iter': 100}

=== Pipe ===

 Pipeline(steps=[('scaler', StandardScaler()), ('kBest', SelectKBest()),

                ('corr\_drop', CorrelationDropper2()),

                ('classifier', KNeighborsClassifier(n\_neighbors=10))])

=== best hyperparametes picked in cv (cv's best score) ===

 {'kBest\_\_score\_func': <function f\_classif at 0x00000237291F1480>, 'kBest\_\_k': 5, 'classifier\_\_n\_neighbors': 10, 'classifier': KNeighborsClassifier(n\_neighbors=10)}

=== feature selection proccess (cv's best score) ===

first feature selection (high corellation to y)

K best selected:

 ['Treatment\_group', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta2\_Fp1', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta2\_Fpz', 'Single\_Mean\_Power\_Abs\_Delta\_Fp1', 'Single\_Mean\_Power\_Abs\_Delta\_Fpz']

=== score (cv's best score): 0.864 +/- 0.074 ===

Fitting 5 folds for each of 100 candidates, totalling 500 fits

=== user arguments ===

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 5, 'n\_iter': 100}

=== Pipe ===

 Pipeline(steps=[('scaler', StandardScaler()), ('kBest', SelectKBest()),

                ('corr\_drop', CorrelationDropper2()),

                ('classifier',

                 SVC(C=0.1, gamma=1, kernel='linear', probability=True,

                     random\_state=42))])

=== best hyperparametes picked in cv (cv's best score) ===

 {'kBest\_\_score\_func': <function mutual\_info\_classif at 0x0000023729210280>, 'kBest\_\_k': 2, 'classifier\_\_kernel': 'linear', 'classifier\_\_gamma': 1, 'classifier\_\_C': 0.1, 'classifier': SVC(C=0.1, gamma=1, kernel='linear', probability=True, random\_state=42)}

=== feature selection proccess (cv's best score) ===

first feature selection (high corellation to y)

K best selected:

 ['Treatment\_group', 'Ratio\_Relative\_Power\_Norm\_Beta1 Over Beta2\_T7']

=== score (CVS's best score): 0.875 +/- 0.063 ===

Fitting 5 folds for each of 100 candidates, totalling 500 fits

=== user arguments ===

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 5, 'n\_iter': 100}

=== Pipe ===

 Pipeline(steps=[('scaler', StandardScaler()), ('kBest', SelectKBest()),

                ('corr\_drop', CorrelationDropper2()),

                ('classifier',

                 DecisionTreeClassifier(max\_leaf\_nodes=2, min\_samples\_split=8,

                                        random\_state=42))])

=== best hyperparametes picked in cv (cv's best score) ===

 {'kBest\_\_score\_func': <function f\_classif at 0x00000237291F1480>, 'kBest\_\_k': 3, 'classifier\_\_min\_samples\_split': 8, 'classifier\_\_max\_leaf\_nodes': 2, 'classifier': DecisionTreeClassifier(max\_leaf\_nodes=2, min\_samples\_split=8, random\_state=42)}

=== feature selection proccess (cv's best score) ===

first feature selection (high corellation to y)

K best selected:

 ['Treatment\_group', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta2\_Fp1', 'Single\_Mean\_Power\_Abs\_Delta\_Fp1']

=== score (cv's best score): 0.875 +/- 0.063 ===

=== user arguments ===

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 5, 'n\_iter': 100, 'n\_jobs': 10}

Fitting 5 folds for each of 100 candidates, totalling 500 fits

=== user arguments ===

  {'rs': 42, 'X\_version': 1, 'split\_rows': 1, 'drop\_out\_correlated': False, 'age\_under\_50': False, 'debug': False, 'classification\_type': 'normal', 'scoring\_method': 'accuracy', 'both': False, 'cv': 5, 'n\_iter': 100, 'n\_jobs': 10}

=== Pipe ===

 Pipeline(steps=[('scaler', StandardScaler()), ('kBest', SelectKBest()),

                ('corr\_drop', CorrelationDropper2()),

                ('classifier',

                 RandomForestClassifier(max\_depth=80, max\_features=21,

                                        min\_samples\_leaf=10,

                                        min\_samples\_split=3, n\_estimators=140,

                                        random\_state=42))])

=== best hyperparametes picked in cv (cv's best score) ===

 {'kBest\_\_score\_func': <function mutual\_info\_classif at 0x0000018F41DC0280>, 'kBest\_\_k': 28, 'classifier\_\_n\_estimators': 140, 'classifier\_\_min\_samples\_split': 3, 'classifier\_\_min\_samples\_leaf': 10, 'classifier\_\_max\_features': 21, 'classifier\_\_max\_depth': 80, 'classifier\_\_bootstrap': True, 'classifier': RandomForestClassifier(max\_depth=80, max\_features=21, min\_samples\_leaf=10,

                       min\_samples\_split=3, n\_estimators=140, random\_state=42)}

=== feature selection proccess (cv's best score) ===

first feature selection (high corellation to y)

K best selected:

 ['Treatment\_group', 'Pair\_Mean\_Power\_Abs\_Delta\_O1-O2', 'Pair\_Mean\_Power\_Abs\_Delta\_T7-T8', 'Pair\_Mean\_Power\_Abs\_Theta\_F7-F8', 'Ratio\_Relative\_Power\_Norm\_Beta1 Over Beta2\_PO4', 'Ratio\_Relative\_Power\_Norm\_Beta1 Over Beta2\_T7', 'Ratio\_Relative\_Power\_Norm\_Delta Over Alpha\_PO3', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta1\_PO8', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta2\_P4', 'Ratio\_Relative\_Power\_Norm\_Delta Over Theta\_F6', 'Ratio\_Relative\_Power\_Norm\_Theta Over Alpha\_FT8', 'Ratio\_Relative\_Power\_Norm\_Theta Over Alpha\_P5', 'Ratio\_Relative\_Power\_Norm\_Theta Over Beta1\_C3', 'Single\_Mean\_Power\_Abs\_Beta1\_FC5', 'Single\_Mean\_Power\_Abs\_Beta1\_Pz', 'Single\_Mean\_Power\_Abs\_Beta1\_T8', 'Single\_Mean\_Power\_Abs\_Beta2\_O2', 'Single\_Mean\_Power\_Abs\_Delta\_FC1', 'Single\_Mean\_Power\_Abs\_Delta\_PO4', 'Single\_Mean\_Power\_Abs\_Delta\_T7', 'Single\_Relative\_Power\_Norm\_Beta1\_C2', 'Single\_Relative\_Power\_Norm\_Beta1\_C3', 'Single\_Relative\_Power\_Norm\_Beta1\_FT8', 'Single\_Relative\_Power\_Norm\_Beta2\_P6', 'Single\_Relative\_Power\_Norm\_Delta\_P5', 'Single\_Relative\_Power\_Norm\_Delta\_Pz', 'Single\_Relative\_Power\_Norm\_Theta\_C2', 'Single\_Relative\_Power\_Norm\_Theta\_CPz']

=== score (cv's best score): 0.875 +/- 0.063 ===

=== Pipe ===

 Pipeline(steps=[('scaler', StandardScaler()), ('kBest', SelectKBest()),

                ('corr\_drop', CorrelationDropper2()),

                ('classifier',

                 <catboost.core.CatBoostClassifier object at 0x0000023D7EF77C70>)])

=== best hyperparametes picked in cv (cv's best score) ===

 {'kBest\_\_k': 10, 'classifier\_\_learning\_rate': 0.1, 'classifier\_\_depth': 10, 'classifier': <catboost.core.CatBoostClassifier object at 0x0000023D7EF77C70>}

=== feature selection proccess (cv's best score) ===

first feature selection (high corellation to y)

K best selected:

 ['Treatment\_group', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta1\_Fp1', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta2\_Fp1', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta2\_Fp2', 'Ratio\_Relative\_Power\_Norm\_Delta Over Beta2\_Fpz', 'Single\_Mean\_Power\_Abs\_Delta\_AF3', 'Single\_Mean\_Power\_Abs\_Delta\_Fp1', 'Single\_Mean\_Power\_Abs\_Delta\_Fpz', 'Single\_Mean\_Power\_Abs\_Delta\_O1', 'Single\_Relative\_Power\_Norm\_Theta\_T7']

=== score (cv's best score): 0.863 +/- 0.078 ===