Perf:

1. baseline classifiers, **no scaling**

<class 'sklearn.ensemble.weight\_boosting.AdaBoostClassifier'>

Model Accuracy: 0.40 (+- 0.14)

Model f1: 0.33 (+- 0.17)

<class 'elm.**ELMClassifie**r'>

Model Accuracy: 0.25 (+- 0.08)

Model f1: 0.21 (+- 0.08)

<class 'sklearn.naive\_bayes.GaussianNB'>

Model Accuracy: 0.46 (+- 0.09)

Model f1: 0.36 (+- 0.13)

<class 'elm.**SimpleELMClassifier'>**

Model Accuracy: 0.46 (+- 0.11)

Model f1: 0.45 (+- 0.11)

<class 'sklearn.neighbors.classification.KNeighborsClassifier'>

Model Accuracy: 0.69 (+- 0.11)

Model f1: 0.68 (+- 0.10)

<class 'sklearn.svm.classes.**LinearSVC'**>

Model Accuracy: 0.42 (+- 0.10)

Model f1: 0.26 (+- 0.16)

<class 'sklearn.ensemble.gradient\_boosting.GradientBoostingClassifier'>

Model Accuracy: 0.72 (+- 0.08)

Model f1: 0.72 (+- 0.08)

<class 'sklearn.linear\_model.stochastic\_**gradient.SGDClassifier'>**

Model Accuracy: 0.15 (+- 0.01)

Model f1: 0.04 (+- 0.02)

<class 'sklearn.ensemble.forest.RandomForestClassifier'>

Model Accuracy: 0.74 (+- 0.08)

Model f1: 0.74 (+- 0.09)

<class 'sklearn.svm.classes**.SVC'>**

Model Accuracy: 0.15 (+- 0.00)

Model f1: 0.04 (+- 0.01)

<class 'sklearn.semi\_supervised.label\_propagation.LabelPropagation'>

Model Accuracy: 0.14 (+- 0.00)

Model f1: 0.04 (+- 0.00)

1. With scaling

Scaler = StandardScaler(with\_mean=False, with\_std=True, copy=False)

MM\_scaler = MinMaxScaler(feature\_range=(0, 15), copy=False)

<class 'sklearn.naive\_bayes.MultinomialNB'>

Model Accuracy: 0.57 (+- 0.14)

Model f1: 0.56 (+- 0.14)

<class 'sklearn.naive\_bayes.BernoulliNB'>

Model Accuracy: 0.58 (+- 0.15)

Model f1: 0.56 (+- 0.16)

***<class 'elm.ELMClassifier'>***

Model Accuracy: 0.48 (+- 0.10)

Model f1: 0.46 (+- 0.10)

**<class 'sklearn.naive\_bayes.GaussianNB'>**

Model Accuracy: 0.43 (+- 0.10)

Model f1: 0.32 (+- 0.13)

<class 'elm.SimpleELMClassifier'>

Model Accuracy: 0.68 (+- 0.10)

Model f1: 0.68 (+- 0.10)

<class 'sklearn.neighbors.classification.KNeighborsClassifier'>

Model Accuracy: 0.69 (+- 0.09)

Model f1: 0.69 (+- 0.09)

<class 'sklearn.svm.classes.LinearSVC'>

Model Accuracy: 0.63 (+- 0.11)

Model f1: 0.61 (+- 0.12)

<class 'sklearn.ensemble.gradient\_boosting.GradientBoostingClassifier'>

Model Accuracy: 0.73 (+- 0.09)

Model f1: 0.72 (+- 0.09)

<class 'sklearn.ensemble.forest.RandomForestClassifier'>

Model Accuracy: 0.75 (+- 0.09)

Model f1: 0.75 (+- 0.10)

2. B) (scale in different order)  
 MM\_scaler = MinMaxScaler(feature\_range=(0, 20))

Scaler = StandardScaler(with\_mean=False, with\_std=True)

X = Scaler.fit\_transform(X.astype(np.float))

X = MM\_scaler.fit\_transform(X.astype(np.float))

<class 'sklearn.naive\_bayes.MultinomialNB'>

Model Accuracy: 0.57 (+- 0.13)

<class 'sklearn.naive\_bayes.BernoulliNB'>

Model Accuracy: 0.58 (+- 0.13)

<class 'elm.SimpleELMClassifier'>

Model Accuracy: 0.68 (+- 0.10)

<class 'sklearn.neighbors.classification.KNeighborsClassifier'>

Model Accuracy: 0.70 (+- 0.10)

<class 'sklearn.svm.classes.LinearSVC'>

Model Accuracy: 0.56 (+- 0.14)

<class 'sklearn.ensemble.gradient\_boosting.GradientBoostingClassifier'>

Model Accuracy: 0.72 (+- 0.08)

<class 'sklearn.ensemble.forest.RandomForestClassifier'>

Model Accuracy: 0.75 (+- 0.10)

1. AdaHERF (selected models, no params set) (+ scaled):

(No prio PCA): mean\_accuracy: 0.81

Added prior whitening PCA: mean\_accuracy: 0.78

LDa – Error

1. PCA + LDA, normal models: (PCA(copy=True, whiten=True); LDA(n\_components=6)

**LDA** models:

<class 'sklearn.naive\_bayes.GaussianNB'>

Model Accuracy: 0.61 (+- 0.07)

<class 'elm.SimpleELMClassifier'>

Model Accuracy: 0.68 (+- 0.11)

<class 'sklearn.neighbors.classification.KNeighborsClassifier'>

Model Accuracy: 0.70 (+- 0.07)

<class 'sklearn.svm.classes.LinearSVC'>

Model Accuracy: 0.61 (+- 0.09)

<class 'sklearn.ensemble.gradient\_boosting.GradientBoostingClassifier'>

Model Accuracy: 0.68 (+- 0.08)

<class 'sklearn.ensemble.forest.RandomForestClassifier'>

Model Accuracy: 0.71 (+- 0.08)

**PCA** models perf:

<class 'sklearn.naive\_bayes.GaussianNB'>

Model Accuracy: 0.51 (+- 0.10)

<class 'elm.SimpleELMClassifier'>

Model Accuracy: 0.68 (+- 0.11)

<class 'sklearn.neighbors.classification.KNeighborsClassifier'>

Model Accuracy: 0.71 (+- 0.09)

<class 'sklearn.svm.classes.LinearSVC'>

Model Accuracy: 0.62 (+- 0.11)

<class 'sklearn.ensemble.gradient\_boosting.GradientBoostingClassifier'>

Model Accuracy: 0.70 (+- 0.11)

<class 'sklearn.ensemble.forest.RandomForestClassifier'>

Model Accuracy: 0.73 (+- 0.09)

# Hyperparams:

**SGD:**

**(n\_iter= 14, loss= hinge, shuffle= True, l1\_ratio= 0.6, penalty= l1, learning\_rate= optimal)**

Mean validation score: **0.61** (std: 0.02)

Parameters: {**'n\_iter': 14, 'loss': 'hinge', 'shuffle': True, 'l1\_ratio': 0.6, 'penalty': 'l1'**, 'learning\_rate': 'optimal'}

Mean validation score: 0.61 (std: 0.02)

Parameters: {**'n\_iter': 7, 'loss': 'modified\_huber', 'shuffle': True, 'l1\_ratio': 0.15, 'penalty': 'l1',** 'learning\_rate': 'optimal'}

**Random Forests:**

**(max\_depth= 17, min\_samples\_split= 2, n\_estimators= 900, criterion= entropy, min\_samples\_leaf= 1)**

**(min\_samples\_leaf= 2, max\_depth= 13, criterion= entropy, min\_samples\_split= 1, n\_estimators= 800)**

RandCV Best params: RandForest

Mean validation score: 0.76 (std: 0.01)

Parameters: {'max\_depth': 17, 'min\_samples\_split': 2, 'n\_estimators': 70, 'criterion': 'entropy', 'min\_samples\_leaf': 1}

Mean validation score: 0.75 (std: 0.01)

Parameters: {'min\_samples\_leaf': 1, 'max\_depth': 13, 'criterion': 'entropy', 'min\_samples\_split': 2, 'n\_estimators': 80}

Mean validation score: 0.75 (std: 0.01)

Parameters: {'min\_samples\_leaf': 2, 'max\_depth': 14, 'criterion': 'entropy', 'min\_samples\_split': 1, 'n\_estimators': 80}

**KNearest Neighbours:**

**KNeighborsClassifier(algorithm= 'kd\_tree',n\_neighbors= 4, p= 1, weights= 'distance'),**

**KNeighborsClassifier(n\_neighbors=3, p=2, weights='distance'),**

Mean validation score: 0.72 (std: 0.01)

Parameters: {'leaf\_size': 30, 'algorithm': 'kd\_tree', 'n\_neighbors': 4, 'p': 1, 'weights': 'distance', 'metric': 'minkowski'}

Mean validation score: 0.72 (std: 0.01)

Parameters: {'weights': 'distance', 'n\_neighbors': **3**, 'algorithm': 'kd\_tree', 'leaf\_size': 30, 'metric': 'minkowski', 'p': **2**}

**GradientBoost Tree**

**(max\_depth= 10, max\_features= auto, subsample= 0.9, min\_samples\_split= 2, min\_samples\_leaf= 2, learning\_rate= 0.14)**

RandomizedSearchCV took 2654.80 seconds for 32 candidates parameter settings.

Mean validation score: **0.76** (std: 0.01)

Parameters: {'max\_depth': 9, 'max\_features': 'auto', 'subsample': 0.9057208535614238, 'min\_samples\_split': 2, 'min\_samples\_leaf': 2, 'learning\_rate': 0.1393099168814611, 'loss': 'deviance'}

Mean validation score: 0.75 (std: 0.01)

Parameters: {'max\_depth': 9, 'min\_samples\_split': 4, 'min\_samples\_leaf': 2, 'max\_features': 'auto', 'subsample': 0.8388784305605327, 'learning\_rate': 0.07384935240725253, 'loss': 'deviance'}

**classic adaboost + PCA**

Starting grid parameter search

RandomizedSearchCV took 757.64 seconds for 4 candidates parameter settings.

Mean validation score: 0.76 (std: 0.01)

Parameters: {'base\_estimator': RandomForestClassifier(bootstrap=True, compute\_importances=None, criterion='gini', max\_depth=11, max\_features=0.6 , min\_samples\_leaf=2,

min\_samples\_split=1, n\_estimators=40), 'n\_estimators': 20}

**classic adaboost best perf**

Mean validation score: 0.77 (std: 0.01)

Parameters: { 'base\_estimator': RandomForestClassifier(bootstrap=True, compute\_importances=None,

criterion='gini', max\_depth=11, max\_features=0.6, min\_samples\_leaf=2,

min\_samples\_split=1, n\_estimators=40, n\_jobs=1), 'n\_estimators': 13}

**ADA-HERf (With submodels modded**), 8 classifiers used + 3 CV:

[SimpleELMClassifier(), GaussianNB(), KNeighborsClassifier(n\_neighbors=3, p=2, weights='distance'), SGDClassifier(n\_iter= 14, loss='hinge', l1\_ratio= 0.6, penalty= 'l1', learning\_rate= 'optimal'), RandomForestClassifier(min\_samples\_leaf= 2, max\_depth= 13, criterion= 'entropy', min\_samples\_split= 1, n\_estimators= 200), GradientBoostingClassifier(max\_depth= 10, max\_features= 'auto', subsample= 0.95, min\_samples\_split= 2, min\_samples\_leaf= 2, learning\_rate= 0.14) ]

**AdaHerf: mean\_accuracy: 0.55**

**AdaHerf + X\_PCA: mean\_accuracy: 0.27**

**RBM + LOGISTIC REGRESSION PARAMETERS**

logistic\_\_C: 10.000000

rbm\_\_learning\_rate: 0.001000

rbm\_\_n\_components: 500.000000

rbm\_\_n\_iter: 50.000000

best score: 0.568

(For - params = {"rbm\_\_learning\_rate": [0.1, 0.01, 0.001], "rbm\_\_n\_iter" : [20, 50],

"rbm\_\_n\_components" : [256,500], "logistic\_\_C" : [1.0, 10.0]})

**RBM + RandomForests PARAMETERS**

best score: 0.695 (Default RF)

rbm\_\_learning\_rate: 0.010000

rbm\_\_n\_components: 500.000000

rbm\_\_n\_iter: 100.000000

**RBM + Tuned RandomForests PARAMETERS**

RF = RandomForestClassifier(min\_samples\_leaf= 2, max\_depth= 13, criterion= 'entropy',

min\_samples\_split= 1, n\_estimators= 800, **max\_features=0.7**)