Report (by Mariya A. Protopova)

Sternberg Dataset

For the Sternberg dataset EEG-data recorded for the study of the relationship between working memory load and motivation was taken. I took the power of the signal in the theta-frequency band (4-8 Hz) from 64 EEG channels that covered the entire head. The features in the dataset are the power of the signal in each time point of the recording, the target was the number of elements to remember (3 – low working memory load, 15 – high working memory load). The target values were transformed to 0 (3 elements) and 1 (15 elements). The idea was to predict the working memory load based on the EEG-signal. The number of observations in each class was approximately equal.

To binarize the data I have split the dataset into train (70%) and test subsets, scaled it by the standard scaler and divided the resulting signals into 4 bins by the quartile. So, each observation was converted to a number in range of 4, reflecting the number of quantile the signal at this moment belonged to. Further, I used one-hot encoding to transform the data from the 4- to 2-valued context. Additionally, I have dropped the first column of each one-hot-encoded category to avoid milticollinearity. Following that from the total number of obtained features I have computed the Fisher information score to select only the most relevant features, to decrease computational load and to increase explanatory power of the models.

Next, I fitted 4 baseline models and a LazyFCA model, using 4 cross-validation iterations and grid-search to find the optimal parameters.

Optimal parameters for the models:

Model	**Optimal Parameters**
KNN	Metric = 'euclidean' n_neighbors = 12 weights = ‘distance’
Logistic Regression	c = 1.1 class_weight = ‘balanced’, penalty = ‘l2’ solver = ‘saga’
XGBoost	objective = 'binary:logistic' eta = 0.01 subsample = 0.1

	threshold > 0.43
Random Forest	class_weight =’balanced’ criterion = ‘gini’ max_depth = 18 min_samples_split = 8
LazyFCA	apha = 0 method = ‘standard’

The resulting models and their quality:

Metric	**KNN**	**Logistic Regression**	**xGBoost**	**Random Forest**	**LazyFCA**
Accuracy	0.668	0.648	0.551	0.712	0.573
F1-measure	0.644	0.622	0.664	0.695	0.636

SC_data dataset

For this data I used magnetoencephalography (MEG) recording from the study where people had to read sentences from the screen. I used time periods, when the participants perceived presented words (target value 1) or saw a blank screen (target value 0). I have picked only those channels that record the signal from the occipital cortex (where the primary visual area is located). The idea was to predict the perceived visual information (word or blank space) based on the MEG recording from the occipital cortex. The features were the signals recorded from the particular channels in the prior selected time-window. The number of observations in each class was approximately equal.

To binarize the data I have split the dataset into train (70%) and test subsets, scaled it by the standard scaler and divided the resulting signals into 6 bins by the quartile. So, each observation was converted to a number in range of 6, reflecting the number of quartiles the signal at this moment belonged to. Further, I used one-hot encoding to transform the data from the 6- to 2-valued context. Additionally, I have dropped the first column of each one-hot-encoded category to avoid milticollinearity. Following that from the total number of obtained features I have computed the Fisher information score to select only

the most relevant features, to decrease computational load and to increase explanatory power of the models.

Next, I fitted 4 baseline models and a LazyFCA model, using 4 cross-validation iterations and grid-search to find the optimal parameters.

Optimal parameters for the models:

Model	**Optimal Parameters**
KNN	Metric = ‘manhattan’ n_neighbors = 29 weights = ‘distance’
Logistic Regression	c = 0.1 class_weight = ‘balanced’, penalty = ‘l1’ solver = ‘saga’
XGBoost	objective = 'binary:logistic' eta = 0.01 subsample = 0.1 threshold > 0.5
Random Forest	class_weight =’balanced’ criterion = ‘gini’ max_depth = 28 min_samples_split =8
LazyFCA	apha = 0 method = ‘standard’

The resulting models and their quality:

Metric	**KNN**	**Logistic Regression**	**xGBoost**	**Random Forest**	**LazyFCA**
Accuracy	0.632	0.620	0.633	0.677	0.634
F1-measure	0.628	0.665	0.654	0.685	0.422 (macro-average)

Aphasia_BCI dataset

For this data EEG recordings of one neurologically healthy person was used. The participant was presented with pictures and had heard corresponding or random words. The person's task was to match (to oneself, but not out loud) a visually presented picture with a pronounced word. The EEG was recorded from 2 channels only located on the forehead. The idea was to predict the cognitive state (target: 1 – matched a word and a picture correctly; 0 – did not match a word and a picture correctly). Each observation in the data set correspond to a time-interval, when a participant was present a word and a picture (an epoch). Features are time points (the number of which reflects the sampling frequency). Due to a comparable number of features and observations I have additionally conducted a PCA to reduce the matrix rank (from 3500 features to 80).

To binarize the data I have split the dataset into train (70%) and test subsets, scaled it by the standard scaler and divided the resulting signals into 4 bins by the quartile. So, each observation was converted to a number in range of 4, reflecting the number of quartiles the signal at this moment belonged to. Further, I used one-hot encoding to transform the data from the 4- to 2-valued context. Additionally, I have dropped the first column of each one-hot-encoded category to avoid milticollinearity. Following that from the total number of obtained features I have computed the Fisher information score to select only the most relevant features, to decrease computational load and to increase explanatory power of the models.

Next, I fitted 4 baseline models and a LazyFCA model, using 4 cross-validation iterations and grid-search to find the optimal parameters.

Optimal parameters for the models:

Model	**Optimal Parameters**
KNN	Metric = 'euclidean' n_neighbors = 4 weights = 'uniform'
Logistic Regression	c = 7.1 class_weight = 'balanced', penalty = 'l2' solver = 'liblinear'
XGBoost	objective = 'binary:logistic' eta = 0.5

	subsample = 0.6 threshold > 0.3
Random Forest	class_weight =’balanced’ criterion = ‘entropy’ max_depth = 8 min_samples_split =13
LazyFCA	apha = 0 method = ‘standard’

The resulting models and their quality:

Metric	**KNN**	**Logistic Regression**	**xGBoost**	**Random Forest**	**LazyFCA**
Accuracy	0.790	0.575	0.696	0.807	0.
F1-measure	0.111	0.261	0.515	0.447	0.