Validation schemes for 2-nd level models

There are a number of ways to validate **second level models** (**meta-models**). In this reading material you will find a description for the most popular ones. If not specified, we assume that the data does not have a time component. We also assume we already validated and fixed hyperparameters for the **first level models (models).**

a) Simple holdout scheme

1. Split train data into three parts: partA and partB and partC.
2. Fit N diverse **models** on partA, predict for partB, partC, test_data getting *meta-features* partB_meta, partC_meta and test_meta respectively.
3. Fit a **metamodel** to a partB_meta while validating its hyperparameters on partC_meta.
4. When the **metamodel** is validated, fit it to [partB_meta, partC_meta] and predict for test_meta.

b) Meta holdout scheme with OOF meta-features

1. Split train data into K folds. Iterate though each fold: retrain N diverse **models** on all folds except current fold, predict for the current fold. After this step for each object in train_data we will have N *meta-features* (also known as *out-of-fold predictions, OOF*). Let's call them train_meta.
2. Fit **models** to whole train data and predict for test data. Let's call these features test_meta.
3. Split train_meta into two parts: train_metaA and train_metaB. Fit a **meta-model** to train_metaA while validating its hyperparameters on train_metaB.
4. When the **meta-model** is validated, fit it to train_meta and predict for test_meta.

c) Meta KFold scheme with OOF meta-features

1. Obtain *OOF predictions* train_meta and test metafeatures test_meta using **b.1** and **b.2**