Cross Validation (model evaluation)

Saturday 27 April 2024 9:38 PM

→ Hold out approach
- shuffle
- train_test_split

problem:

→ Variability; the performance of the model can be very sensitive to how the data is divided into training & testing sets. If the split is unfortunate, the training set may not be representative of the overall distribution of data. This leads to high variance in the estimation of the model's performance (unreliable metrics) ↑ variance

→ The holdout approach only uses a portion of the data for training & a different portion for testing. This means that the model doesn't get to learn from all available data. problematic if dataset is small.

→ If some classes or patterns are over- or under-represented in the training set or the test set due to the random split, it can lead to a biased performance estimation. [Doesn't see the big picture] ↑ bias

→ If holdout method is used for hyperparameter tuning, there's a risk of overfitting to the test set b/c info might leak from the test set into the model. Model's performance on the test set might be overly optimistic & not representative of its performance on unseen data. (data leakage)

→ Resampling
- Permutation tests
- Bootstrap
- Cross validation

→ The idea of cross-validation is to divide the data into several subsets or folds. The model is then trained on some of these subsets & tested, on the remaining ones. This process is repeated multiple times, with different subsets used for training & validation each time.

→ The results from each round are usually averaged to estimate the model's overall performance.

→ Cross validation
- time based
- Leave one out CV
- Leave p out CV
- K-fold CV

Leave p out

→ Cross validation
- K-fold CV
- stratified
- nested k fold
- repeated k fold

→ Find TRUE model performance on unseen data

→ Anchoring Randomization (Random state), will obtain same results when run multiple times. Can also use for benchmarking.