Hyperparameter Tuning P1

ENSF 444

Review Cross validation

Discuss how to search for the best hyperparameters

Review Cross Validation

What is cross validation?

a technique for evaluating the performance of machine learning model on different sub sets of the data.

What does it do?
Helps to avoid overfitting and under-fitting AND also estimate the generalization error of the model.

Common Cross Validation - K-fold Cross validation where data is split into k equal parts / folds

`K-fold cross-validation is a popular technique used to assess the performance of a machine learning model and to mitigate issues related to bias and variance in the model evaluation process. It involves partitioning the original dataset into K equal-sized subsets (or "folds"), using K-1 folds for training the model and the remaining fold for testing. This process is repeated K times, with each fold serving as the test set exactly once.

Here's how K-fold cross-validation works:

1. **Partitioning the Data:** The original dataset is randomly partitioned into K equal-sized subsets, or folds.
2. **Model Training and Evaluation:** For each iteration:
    - One of the K folds is set aside as the test set.
    - The model is trained on the remaining K-1 folds (i.e., the training set).
    - The trained model is then evaluated on the held-out test fold to compute a performance metric (e.g., accuracy, F1 score).

3. **Aggregating Results:** After K iterations, you will have K different performance scores (one for each fold). These scores are typically averaged to obtain a single performance estimate for the model.

Here are the advantages of K-fold

- **Better Performance Estimation:** By repeating the process K times with different subsets for training and testing, K-fold cross-validation provides a more robust estimate of model performance compared to a single train-test split. It helps to reduce the variability in performance estimation and provides a more accurate assessment of how well the model generalizes to unseen data.
- **Maximizing Data Utility:** K-fold cross-validation ensures that every data point is used for both training and testing at least once. This maximizes the utilization of the available data and provides a more comprehensive evaluation of the model's performance.
- **Model Selection:** K-fold cross-validation can be used for hyperparameter tuning and model selection. By evaluating different models or hyperparameters on multiple folds, you can choose the model or configuration that performs the best on average across the folds

Cross validation allows us to compare different machine learning methods and get a sense of how well they work in practice

Cross validation uses all data at a time and summarizes it at the end

Choosing a machine learning method after the cross validation is better than to choose a ML first.

Because Cross validation will help compare the models

https://www.youtube.com/watch?v=fSytzGwwBVw&ab_channel=StatQuestwithJoshStarmer

Helpful video to understand more of Cross validation

Basically in this example, it did a 5 fold test, where each fold is split into 80/20 for training/test. And it will test it on different parts of the data in a 80/20 style.

Fold	Train R^2 Score	Test R^2 Score	Train MSE Score	Test MSE Score
1	0.984939	0.915023	0.001726	0.010029
2	0.985390	0.878435	0.001702	0.013382
3	0.986641	0.850011	0.001588	0.014624
4	0.984555	0.906993	0.001766	0.011061
5	0.983009	0.927178	0.001905	0.009316

We are given the Standard Error equation for some reason

Standard Error

UNIVERSITY OF CALGARY

The Standard Error (SE) is a way to measure how accurate an estimated value is, like the coefficients in a linear regression. It shows the average amount by which estimates can vary when we take multiple samples from the same group.

The Standard Error (SE) is typically calculated using this formula:

$$\text{SE} = \frac{s}{\sqrt{n}}$$

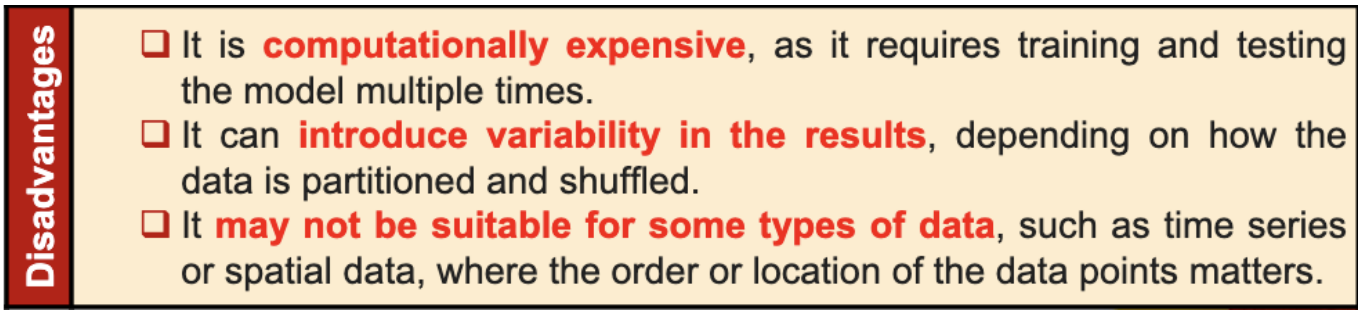
where:

- s represents the sample standard deviation, which measures the variability of data points within the sample.
- n is the number of data points (sample size).

Advantages/Disadvantages of Cross Validation and

Advantages

- ✓ It helps to **prevent overfitting**, which occurs when a model is trained too well on the training data and performs poorly on new, unseen data.
- ✓ It provides a **more realistic estimate** of the model's generalization performance, i.e., its ability to perform well on new, unseen data.
- ✓ It can be used to **compare different models** and select the one that performs the best on average.



Stratified Cross-Validation

Method of evaluating the performance of a machine learning model on **UNSEEN DATA**

Similar to Cross Validation but it ensures that each fold or subset of data has the same proportion of classes as the original

Especially useful for classification problems where the outcome variable has tow or more categories

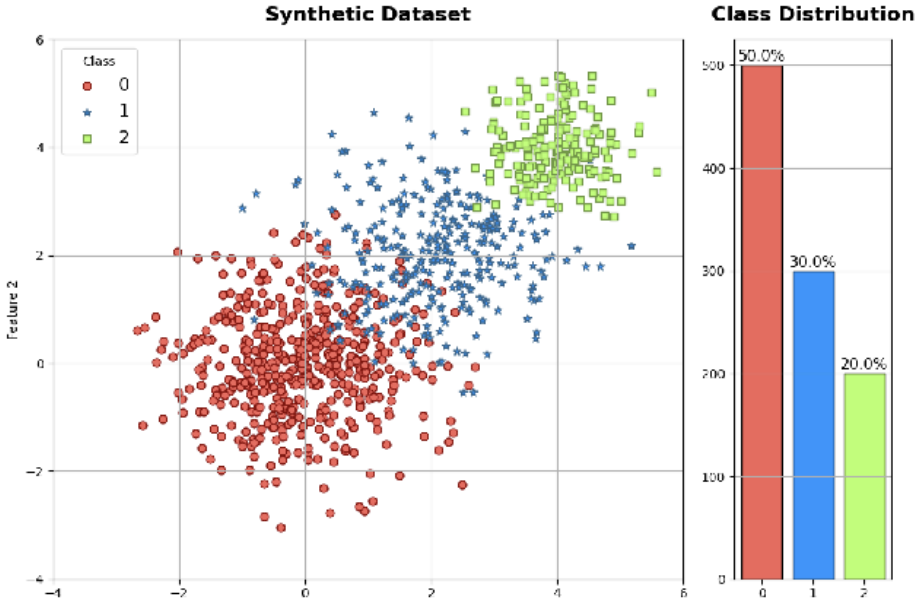
An example, if we want to predict whether a person has a disease or not, we need to make sure that the training and test sets have the same ration of positive and negative cases, if not it will overfit or underfit the data

The difference between Regular and Stratified

1. **Stratified Validation:**
    - **Purpose:** Stratified validation is a technique used to ensure that the distribution of classes or categories in the dataset is maintained across the training and validation sets, especially when dealing with imbalanced datasets.
    - **Usage:** Typically used in scenarios where there is a significant class imbalance, meaning that some classes have much fewer instances than others.
    - **Process:** The dataset is divided into training and validation sets in a way that preserves the proportion of different classes in each set. For example, if there are three classes in the dataset and class A comprises 20% of the data, class B 30%, and class C 50%, then the training and validation sets will also have approximately the same class distribution.
2. **Cross-Validation:**
    - **Purpose:** Cross-validation is a technique used to assess the performance of a machine learning model by partitioning the dataset into multiple subsets and training the model multiple times, using different subsets as training and validation sets.

- **Usage:** Widely used for evaluating model performance, selecting hyperparameters, and assessing the generalization ability of the model.
- **Process:** The dataset is divided into K equal-sized folds. The model is trained K times, each time using a different fold as the validation set and the remaining folds as the training set. The performance scores obtained from each fold are then averaged to obtain a final performance estimate.

```
# Generate synthetic data
X, y = make_blobs(n_samples = [500, 300, 200], centers=[[0, 0], [2, 2], [4, 4]],
                  n_features=2, random_state=0, cluster_std=[1.0, 1, .6])
```



Here is data for stratified

- **Class Proportions (in bothTrain and Test):**
  - **Class 0:** 50%
  - **Class 1:** 30%
  - **Class 2:** 20%
- # of rows in the **Train Set = 800**
- # of rows in the **Test Set = 200**
- These are the same distribution as the original dataset.

Fold	Train Accuracy Score	Test Accuracy Score	Train F1 Score (weighted)	Test F1 Score (weighted)
1	0.99375	0.925	0.993747	0.924887
2	0.99625	0.920	0.996252	0.920353
3	0.99625	0.925	0.996248	0.925667
4	0.99375	0.930	0.993758	0.928942
5	0.99000	0.935	0.989983	0.934889

Heres how you would split it

Reducing Overfitting

(for gradient boosting classifiers)

1. Limiting the number of trees in the model. More trees increase the complexity and variance of the model. Use 'n_estimators' to control the number of trees
2. Applying learning rates
3. Tuning the hyperparameters - max_depth, min_samples_split, min_samples_leaf, max_features and gamma
4. Using **GridSearch**