K-Fold Cross-Validation

Objectives

In this reading, you will explore various aspects of cross-validation, including:

- **Introduction to cross-validation and K-Fold cross-validation**
- **Types of K-Fold cross-validation**
- **Comprehending model complexity in linear regression through K-Fold cross-validation**

Introduction to cross-validation and K-Fold cross-validation

Cross-validation is a statistical technique employed in machine learning to evaluate the performance and generalizability of a predictive model. The primary objective of cross-validation is to divide a data set into subsets, train the model on some of these subsets, and assess its performance on the remaining subset. We repeat this process multiple times and average the model's performance metrics over these iterations.

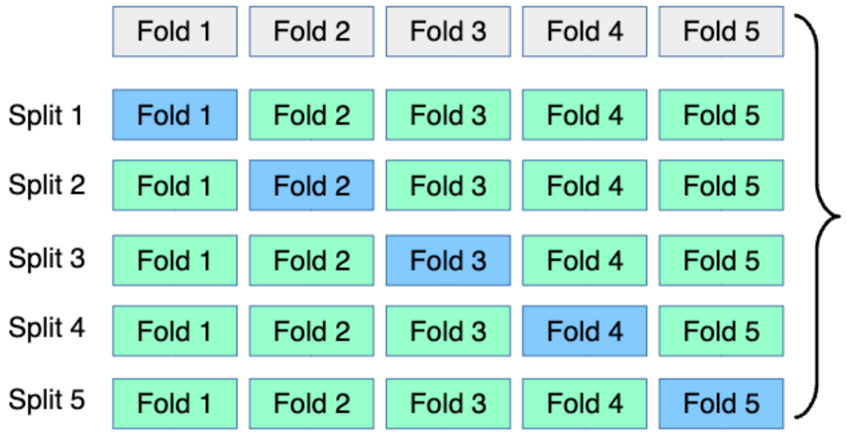
Cross-validation offers several advantages, including more reliable performance evaluation by reducing the impact of variability in a single random data split. It also optimizes the utilization of available data, ensuring that each data point is used for validation exactly once.
The most prevalent form of cross-validation is **k-fold** cross-validation.

In k-fold cross-validation, the dataset is randomly split into k equally sized folds or subsets.

Subsequently, the model is trained on **k-1** folds and validated on the remaining fold. This cycle is iterated k times, with each iteration employing a unique fold as the test set.
The final performance metric is typically the average of the metrics obtained in each iteration.

For instance, if you choose k = 5, you create five folds. In this approach, you train your model 5 times. In each iteration, a different fold is the validation set, and you train your model on the other four-folds.
Consequently, every data point can be part of the validation set precisely once.

The figure provided depicts this process:



source scikit-learn.org

Types of K-Fold cross-validation

Standard K-Fold cross-validation

In traditional K-Fold cross-validation, the data set undergoes random partitioning into K folds of roughly equal size. In each iteration, one fold is the validation set, while the remaining K - 1 folds constitute the training set. We repeat this procedure K times, ensuring that each fold is used exactly once as the validation set.

Stratified K-Fold cross-validation

Stratified K-Fold cross-validation aims to maintain a consistent distribution of classes in each fold, aligning with the overall proportion observed in the data set. This process proves particularly valuable for data sets with imbalances where certain classes are scarce. The goal is to ensure unbiased model evaluation by accurately representing the class distribution in each fold.

While the data selection is not entirely random, there is an element of randomness in choosing samples within each class.

For example, consider a binary classification scenario with an imbalanced data set, where 90% of samples belong to **Class A** and 10% to **Class B**. In stratified 5-fold cross-validation, each fold will include a proportionate representation of **Class A** and **Class B** samples, consistently preserving the imbalance ratio across folds.

Group K-Fold cross-validation

Group K-Fold cross-validation is employed when working with data sets where samples exhibit interdependencies, such as time-series data or data with spatial correlations. This method guarantees that samples from the same group are either entirely within the training or validation set, preventing data leakage across folds.

In contrast to certain other cross-validation techniques, we do not randomly divide the data. Instead, we utilize the grouping information to construct folds that preserve the cohesion of the groups.

As an illustration, consider a data set comprising multiple time-series sequences, each representing data collected over time from an individual. With group K-fold cross-validation, the sequences (groups) remain intact across folds, ensuring the model avoids learning from future data when making predictions.

Comprehending model complexity in linear regression through K-Fold cross-validation

Based on the complexity, we can categorize machine learning models as follows:

- **Low complexity models**
- **High complexity models**

Low complexity models: These are relatively simple models trained and built with a limited number of features. They are easy to understand and interpret and work well with small data sets.
These models cannot capture complex relationships and are not flexible in adapting to diverse data patterns.
These are prone to underfitting, meaning they do not provide good accuracy on the training and test data.

High complexity models: These are sophisticated models that are built and trained with more features. They are more flexible when adapting to diverse patterns of data. This flexibility may also cause overfitting, where the model may memorize noise or specific patterns that do not generalize well when testing data.

K-fold cross-validation is a powerful technique used to evaluate the performance of machine learning models.
By partitioning the data set into K subsets or folds and systematically rotating them as training and validation sets, K-fold cross-validation offers a more resilient assessment of the model's performance.

Here are some general observations:

Smaller k (k = 2 or 3)
By selecting smaller values for k in K-fold cross-validation, each fold encompasses a more significant portion of the data set, providing the model with more extensive training data. This results in a reduced variance in parameter estimates as the model benefits from more data for learning.
Low variance, in this context, implies minimal fluctuation in performance metrics (such as mean squared error) across different folds. The model consistently performs well, irrespective of the specific subset of data used for training and validation. This effective variance management helps mitigate or nearly eliminate **overfitting**, ensuring the model's stability and reliability.

Larger k (k = 10 or 20)
As k increases in K-fold cross-validation, the model trains on reduced subsets as each fold encompasses a smaller portion of the dataset. This circumstance can lead to higher variance in parameter estimates due to the model having less data to learn from in each iteration.

High variance, in this context, denotes significant fluctuations in performance metrics across different folds. The model may exhibit exceptional performance on some folds but poorly on others. This pattern suggests that the model is overfitted to the training data and lacks generalization capability, yielding a highly complex model.

To illustrate high variance, consider a real-time example of building a regression model to predict house prices. The objective is to predict house prices based on square footage, number of bedrooms, and neighborhood. In two different subsets of data representing distinct folds in cross-validation:

- **Fold 1:** We train the model on a subset where houses with large square footage and many bedrooms are prevalent. The learned model may prioritize these features as crucial for predicting house prices.

- **Fold 2:** We then train the model on a different subset where houses in a diverse neighborhood, regardless of square footage, tend to have higher prices. In this case, the learned model assigns more importance to the neighborhood than in Fold 1.

Therefore, these two subsets demonstrate considerable variance.

Conclusion

This reading mainly overviews K-fold cross-validation and its significance in evaluating a model's ability to generalize to unseen data. Importantly, it's worth noting that the number of folds in K-fold validation does not inherently result in underfitting.

Congratulations! You have just completed this reading on K-Fold Cross-Validation.

Author(s)

Geetika Pal

Lakshmi Holla

Other Contributors

Malika Singla