GJU
الجامعة الألمانية الأردنية
German Jordanian University

SEEIT

Cross Validation

Rana Husni

GJU
الجامعة الألمانية الأردنية
German Jordanian University

SEEIT

Cross Validation in Machine Learning

In machine learning, we couldn't fit the model on the training data and can't say that the model will work accurately for the real data. For this, we must assure that our model got the correct patterns from the data, and it is not getting up too much noise. For this purpose, we use the cross-validation technique.

What is Cross-Validation?

Cross validation is a technique used in machine learning to evaluate the performance of a model on unseen data. It involves dividing the available data into multiple folds or subsets, using one of these folds as a validation set, and training the model on the remaining folds. This process is repeated multiple times, each time using a different fold as the validation set. Finally, the results from each validation step are averaged to produce a more robust estimate of the model's performance. Cross validation is an important step in the machine learning process and helps to ensure that the model selected for deployment is robust and generalizes well to new data.

What is cross-validation used for?

The main purpose of cross validation is to prevent overfitting, which occurs when a model is trained too well on the training data and performs poorly on new, unseen data. By evaluating the model on multiple validation sets, cross validation provides a more realistic estimate of the model's generalization performance, i.e., its ability to perform well on new, unseen data.

Rana Husni

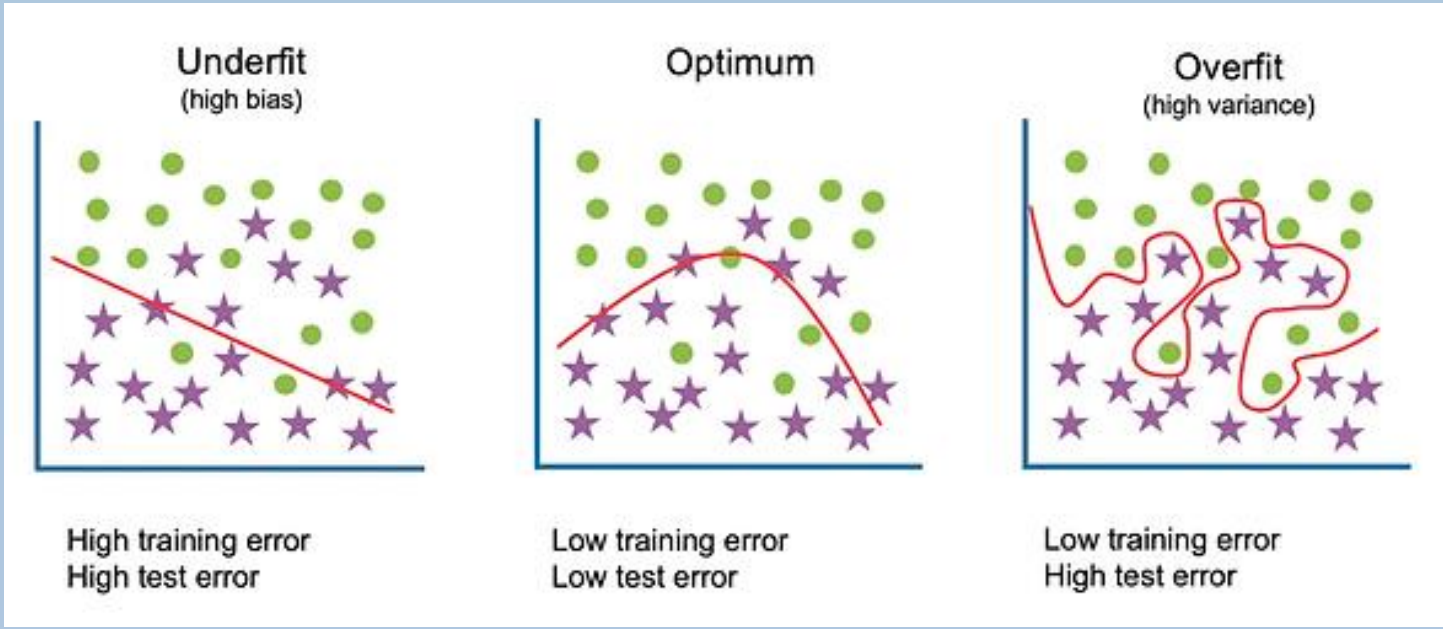
SEEIT

Any machine learning model needs to always predict the correct output across a variety of different input values, present in diverse datasets. This aspect of a machine learning model is called **stability**. If a model does not vary much when the input data is altered, it means that it has been trained well to generalize and find patterns in our data. A model can lose stability in two ways:

Underfitting: It arises when the model does not fit properly with training data. It does not discover patterns in the data and thus when it is given new data to predict, it cannot find patterns in it too. It underperforms on both known and unrecognized data.

Overfitting: When the model trains nicely on training data and generalizes to it, but fails to serve on new, unrecognized data. It grabs slight variations in training data and cannot serve on data that does not have the same variations.

The figures shown below show unfit, overfit, and optimally fit models:

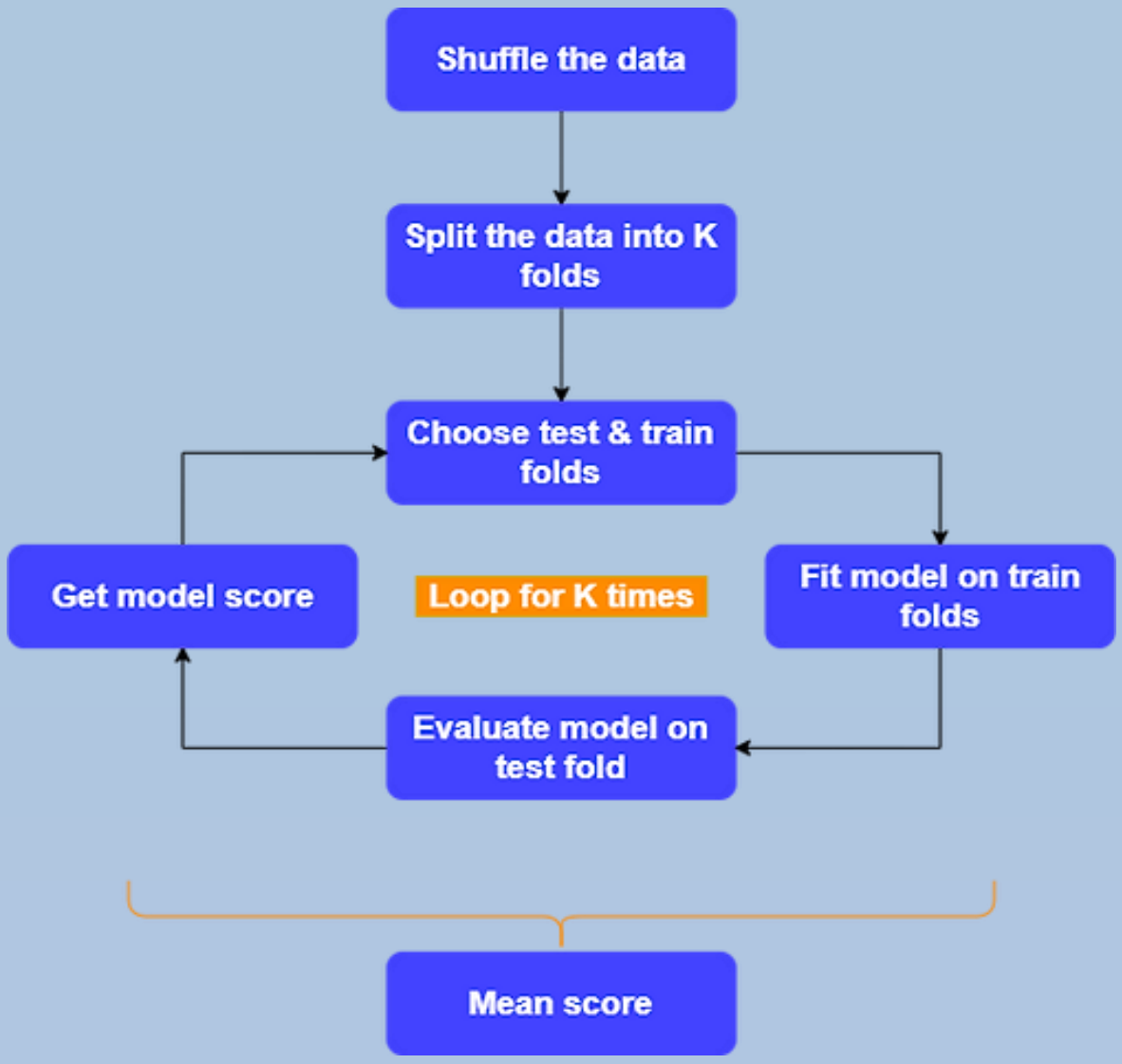


In Figure 1, we can see that the model does not completely capture all the features of our data and ditches out some important data points. This model has generalized our data too much and is under-fitted.

In figure 2, the model captures the intricacies of our model while ignoring the noise, this model is our optimal model.

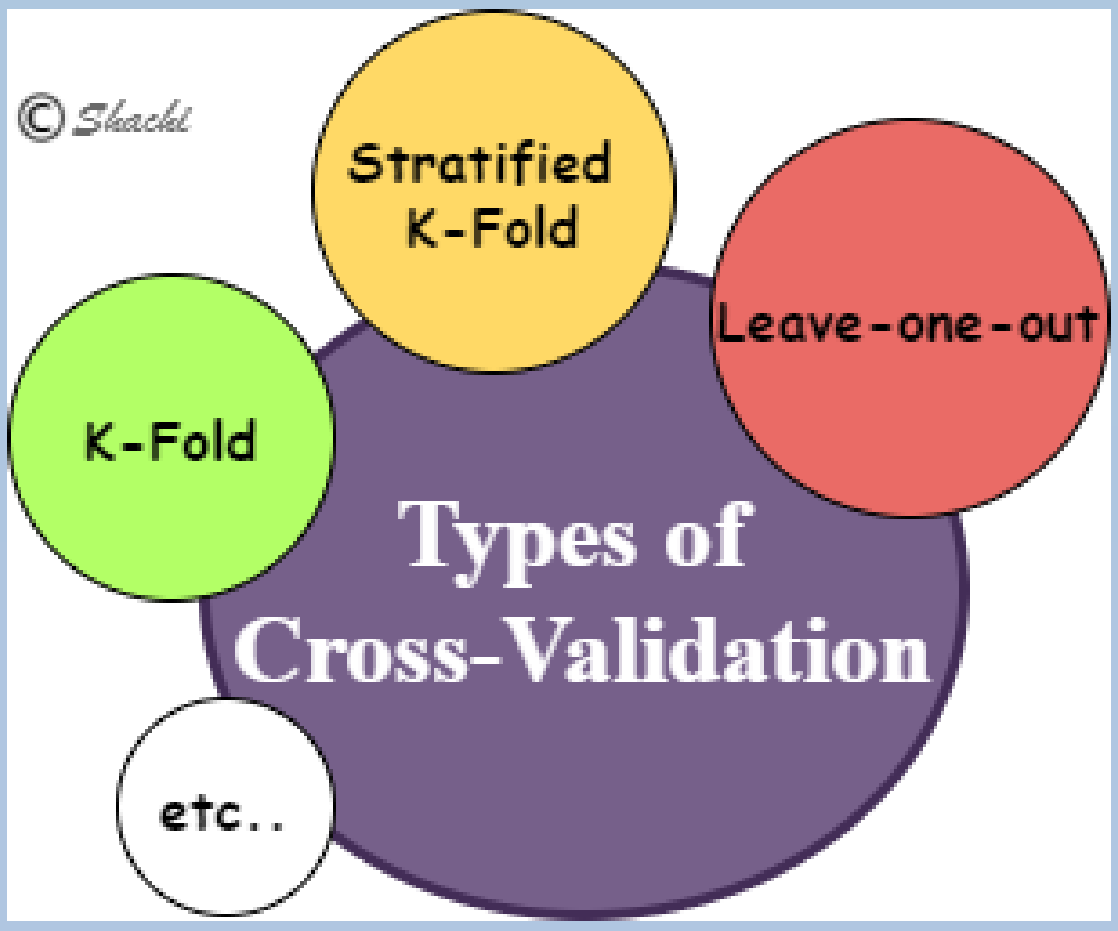
In Figure 3, our model has captured every single characteristic of the data, including the noise. If we were to give it a different dataset, it would not be able to predict it as it is too explicit to our training data, hence it is overfitted.

Cross-Validation



Types of Cross-Validation

There are several types of cross validation techniques, including k-fold cross validation, leave-one-out cross validation, and Holdout validation, Stratified Cross-Validation. The choice of technique depends on the size and nature of the data, as well as the specific requirements of the modeling problem.



GJU
الجامعة الألمانية الأردنية
German Jordanian University

K-Fold Cross Validation

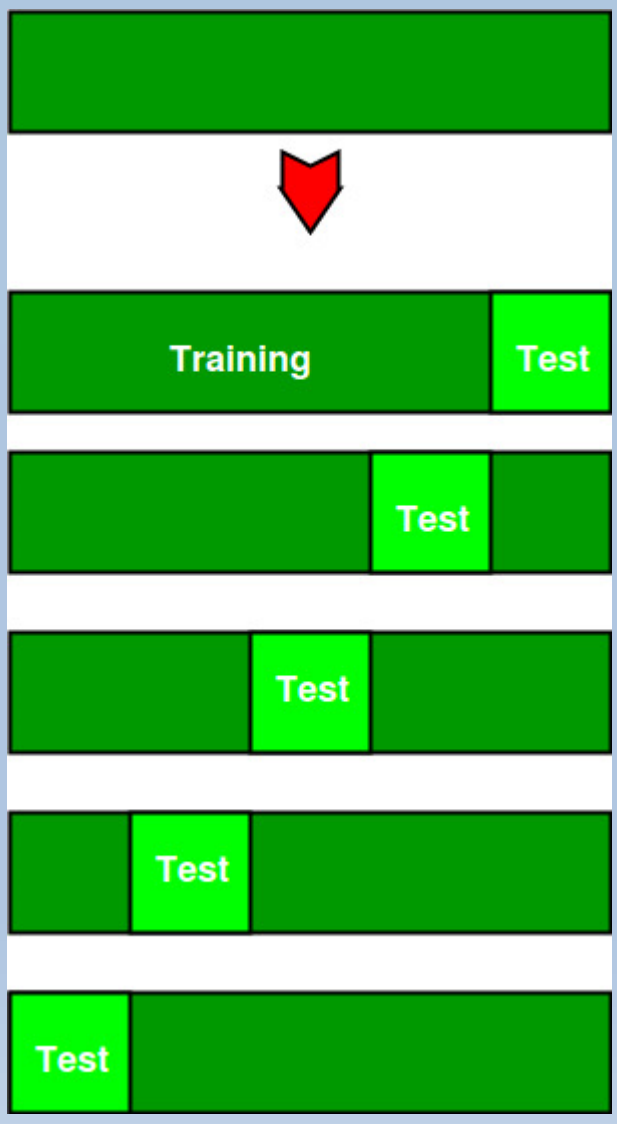
SEEIT

In K-Fold Cross Validation, we split the dataset into k number of subsets (known as folds) then we perform training on the all the subsets but leave one(k-1) subset for the evaluation of the trained model. In this method, we iterate k times with a different subset reserved for testing purpose each time.

Note: It is always suggested that the value of k should be 10 as the lower value of k is takes towards validation and higher value of k leads to LOOCV method.

Leave-one-out cross validation (LOOCV) is a type of cross-validation method in which a single data point is removed from the dataset, and the model is trained on the remaining data points. The removed data point is then used as a test case to evaluate the model's performance.

The diagram shows an example of the training subsets and evaluation subsets generated in k-fold cross-validation. Here, we have total 25 instances. In first iteration we use the first 20 percent of data for evaluation, and the remaining 80 percent for training ([1-5] testing and [5-25] training) while in the second iteration we use the second subset of 20 percent for evaluation, and the remaining three subsets of the data for training ([5-10] testing and [1-5 and 10-25] training), and so on.



Rana Husni

K-folds cross-validation

SEEIT

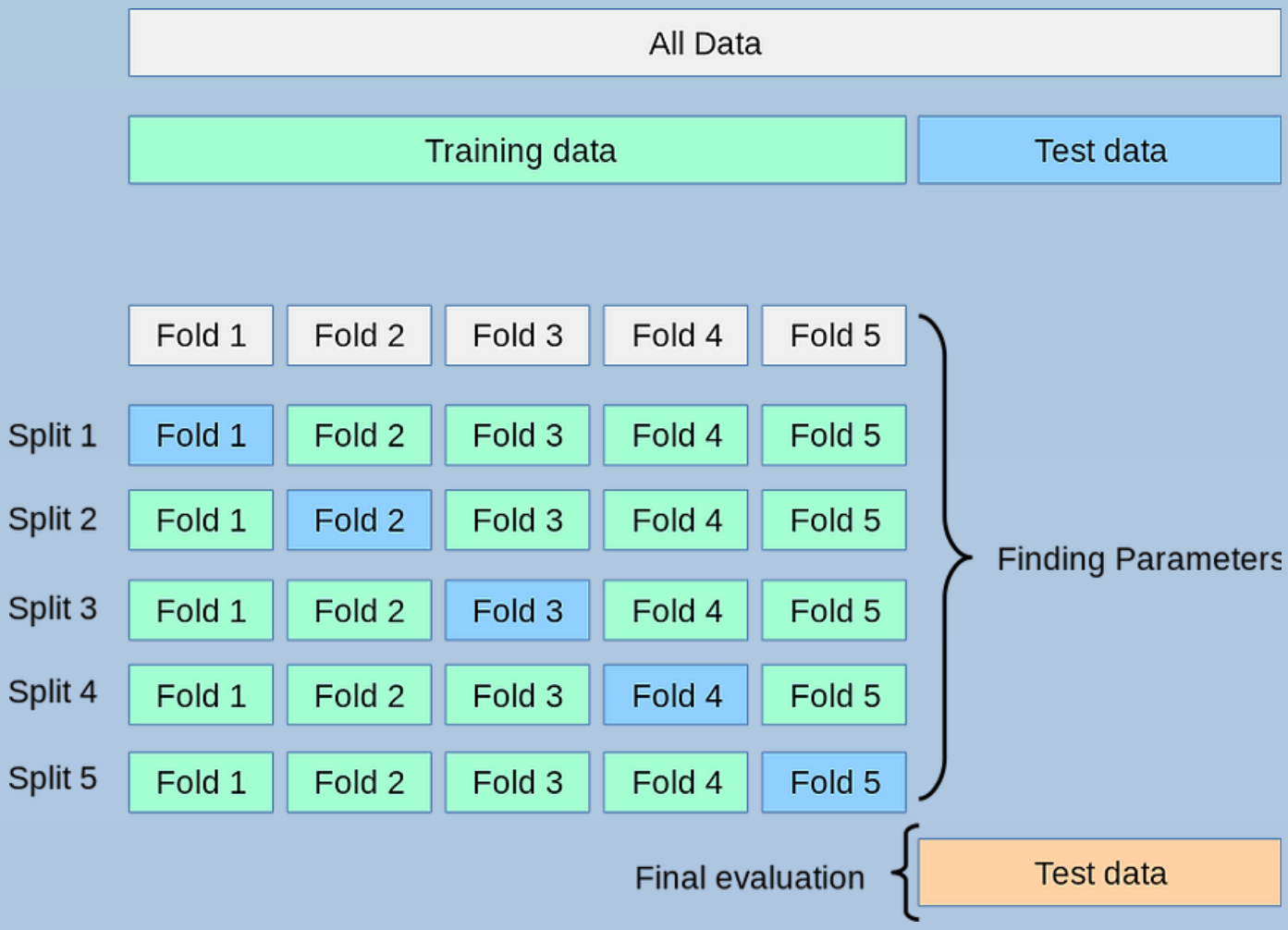
In the case of K-Fold cross-validation input data is divided into ‘K’ number of folds, hence the name K Fold. Consider we have divided data into 5 folds i.e. K=5. Now we have 5 sets of data to train and test our model. So the model will get trained and tested 5 times, but for every iteration, we will use one fold as test data and the rest all as training data. Note that for every iteration, data in training and test fold changes which adds to the efficacy of this method.

This remarkably decreases underfitting as we are using most of the data for training(fitting), and most of the data is also being used in the validation set. K Fold cross-validation aids to generalize the machine learning model, which results in better predictions on unexplored data.

GJU
الجامعة الألمانية الأردنية
German Jordanian University

SEEIT

K-folds cross-validation



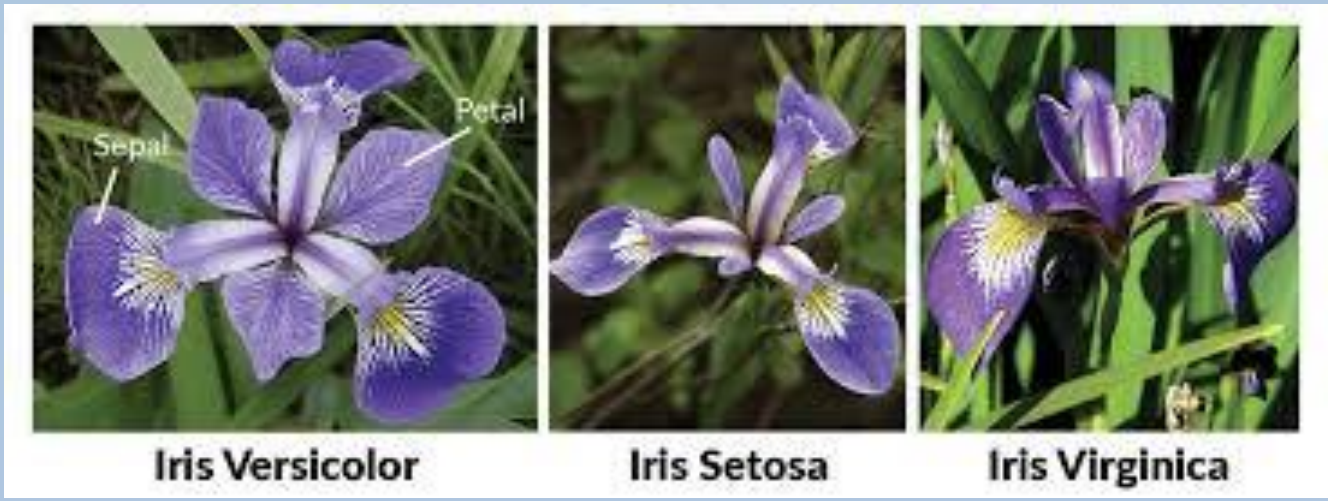
Total instances: 25
Value of k : 5

No. Iteration	Training set observations	Testing set observations
1	[5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]	[0 1 2 3 4]
2	[0 1 2 3 4 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]	[5 6 7 8 9]
3	[0 1 2 3 4 5 6 7 8 9 15 16 17 18 19 20 21 22 23 24]	[10 11 12 13 14]
4	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 20 21 22 23 24]	[15 16 17 18 19]
5	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19]	[20 21 22 23 24]

Rana Husni

Iris dataset

The Iris dataset is a classic benchmark dataset in machine learning and statistics. It consists of 150 samples of iris flowers, each belonging to one of three species: setosa, versicolor, or virginica. For each sample, four features are measured: the length and width of the sepals and petals, in centimeters.



Rana Husni

KFold


```
from sklearn import datasets
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score,KFold


  X, y = datasets.load_iris(return_X_y=True)

  clf = DecisionTreeClassifier(random_state=42)

  k_folds = KFold(n_splits = 5)

  scores = cross_val_score(clf, X, y, cv = k_folds)

  print("Cross Validation Scores: ", scores)
  print("Average CV Score: ", scores.mean())
  print("Number of CV Scores used in Average: ", len(scores))
```

```
Cross Validation Scores:  [1.         1.         0.83333333 0.93333333 0.8       ]
Average CV Score:  0.9133333333333333
Number of CV Scores used in Average:  5
```

GJU
الجامعة الألمانية الأردنية
German Jordanian University

SEEIT

Stratified K-Fold

In cases where classes are imbalanced we need a way to account for the imbalance in both the train and validation sets. To do so we can stratify the target classes, meaning that both sets will have an equal proportion of all classes.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import StratifiedKFold, cross_val_score
from sklearn.datasets import load_breast_cancer
data = load_breast_cancer()
X=data.data
y= data.target
from collections import Counter

print("Original class distribution:", Counter(y))

clf = DecisionTreeClassifier(random_state=42)

sk_folds = StratifiedKFold(n_splits = 5)

scores = cross_val_score(clf, X, y, cv = sk_folds)

print("Cross Validation Scores: ", scores)
print("Average CV Score: ", scores.mean())
print("Number of CV Scores used in Average: ", len(scores))
```

```
Original class distribution: Counter({1: 357, 0: 212})
Cross Validation Scores:  [0.9122807  0.90350877 0.92982456 0.95614035 0.88495575]
Average CV Score:  0.9173420276354604
Number of CV Scores used in Average:  5
```

Rana Husni