29.14 k'-Fold Cross-Validation

So far we have seen the dataset 'D' being split into 3 parts. They are D_{Train}(60%), D_{cv}(20%) and D_{Test}(20%). We have been using 'D_{Train}' to find out the nearest neighbors and 'D_{cv}' to find out the optimal 'K' value, but because of this, after obtaining the optimal 'K', we are training the final optimal model on 'D_{Train}' and making predictions on 'D_{Test}'. The 'D_{cv}' is not at all used anywhere after getting the optimal 'K'. So it goes to waste.

In order to make the maximum usage of our data and not letting any subset of data go to waste, we have come up with a strategy called K-fold Cross-Validation. For this, we have to split the dataset 'D' only into two parts. They are 'D_{Train}'(80%) and 'D_{Test}'(20%). The 'D_{cv}' gets created internally during the cross-validation. More the data used for the training, the more the model's predictive power would be. But we cannot skip the test data(D_{Test}). So the 'D_{cv}' would be created from D_{Train}.

Procedure for k'-Fold Cross-Validation

1) The dataset 'D' is split into 'D_{Train}' and 'D_{Test}'.
   D_{Train} → Finding out the Nearest Neighbors and obtaining the optimal 'K'(hyperparameter of K-NN) value
   D_{Test} → Unseen data on which the final model has to run and compute the accuracy.
2) The training data (D_{Train}) is divided into k' parts(let us assume k'=4 for now).
   So the 4 parts would be 'D_1', 'D_2', 'D_3' and 'D_4'.

D_{Train}

D_1	**D_2**	**D_3**	**D_4**

In k'-fold CV, the number of parts into which 'D_{Train}' is divided is equal to **k'**.

3)

	D_{Train}	**D_{cv}**	**Accuracy on D_{cv}**
K=1	**$D_1D_2D_3$**	**D_4**	**$a_4^{(1)}$**
K=1	**$D_1D_2D_4$**	**D_3**	**$a_3^{(1)}$**
K=1	**$D_1D_3D_4$**	**D_2**	**$a_2^{(1)}$**
K=1	**$D_2D_3D_4$**	**D_1**	**$a_1^{(1)}$**
K=2	**$D_1D_2D_3$**	**D_4**	**$a_4^{(2)}$**
K=2	**$D_1D_2D_4$**	**D_3**	**$a_3^{(2)}$**
K=2	**$D_1D_3D_4$**	**D_2**	**$a_2^{(2}$)**
K=2	**$D_2D_3D_4$**	**D_1**	**$a_1^{(2)}$**
.	.	.	.

Avg. Accuracy Score for K=1 → $\mathbf{(a_1^{(1)}+a_2^{(1)}+a_3^{(1)}+a_4^{(1)})/4}$. Let us denote this as '$\mathbf{a_{k=1}}$'.

Avg. Accuracy Score for K=2 → $\mathbf{(a_1^{(2)}+a_2^{(2)}+a_3^{(2)}+a_4^{(2)})/4}$. Let us denote this as '$\mathbf{a_{k=2}}$'. Like this, we have to compute the average CV scores for all the values of 'K'. (Here 'K' is the hyperparameter in K-NN)

4) We should now plot a 2D line plot, with the 'K'(hyperparameter in K-NN) values on the 'X' axis, and their corresponding average CV accuracies on the 'Y' axis.

   Whichever value of 'K' gives the highest average CV accuracy score, that would be considered as the optimal 'K' value.
5) Fit the final K-NN models using the obtained optimal 'K' value on the whole 'D_{Train}', and make predictions on 'D_{Test}'(unseen data), and compute the final test accuracy score.

Note: In k'-fold cross-validation, the training data 'D_{Train}' is divided into k' folds and each fold is used in both the training and the cross-validation phases. The most typically used values for k' are 3 (or) 5.

K'-fold cross-validation gives the optimal value of the hyperparameter such that the results obtained on the unseen data are more generalized.