Important Notes

Cross-Validation: Cross-validation is a step in the process of building a machine learning model which helps us ensure that our models fit the data accurately and also ensures that we do not **overfit**.

Overfitting: When our machine learning model continues to fit perfectly on the training set and performs poorly on the test set, we then say our model is overfitting. Another definition of overfitting would be when the test loss increases as we keep improving training loss.

Different methods of doing cross-validation:

(a) k-fold cross-validation (b) stratified k-fold cross-validation
(c) hold-out based validation (d) leave-one-out cross validation
↳ a lot of computational power is required, no one uses it now.
(e) group k-fold cross-validation.

Choosing appropriate cross-validation strategy depends on the type of data

→ In **k-fold cross-validation**, we divide the dataset into k-different sets which are exclusive of each other. This strategy can be used with almost any kind of dataset.

→ The **stratified k-fold cross-validation**, keeps the ratio of labels in each fold constant, then we can use this cross-validation strategy in class imbalance datasets.

* The rule is simple if it's a standard classification problem choose stratified k-fold blindly.

→ **Hold-out based cross-validation**, In this we keep a certain set of samples on hold for cross-validation. It is usually used in case of large datasets and time-series forecasting.

Important Notes

→ When we have really small datasets, creating big validation sets is not possible because we will be left with very less amount of data for training the model. In those cases, we can opt for a type of k-fold cross-validation where k = N, where N is the number of samples in the dataset. This means in all folds of training, we will be training on all data samples except 1. The no. of folds for this type of cross-validation will be same as the no. of samples we have in the dataset.

For regression problems –

* We can use all the cross-validation techniques that we mentioned earlier, except for stratified k-fold cross-validation.

→ The stratified k-fold cross-validation must be applied in situation where we see that distribution of targets is not consistent. To use stratified k-fold, we first need to divide the target into bins and then we can use stratified k-fold as we do in case of classification problems.

How to divide samples into bins?

(a) When we have a lot of samples (>10k, >100k), then we can divide the dataset into 10 – 20 bins.

(b) When we have less no. of samples, we use the **Sturge's** rule for binning the samples, where no. of bins is given by $\boxed{\text{No. of bins} = 1 + \log_2(N)}$, samples

Conclusion: Cross-validation is the first and most essential step when it comes to building ML models. If you want to do feature engineering, split your data first, If you are going to build ML models ~~first~~ split your data first.

Important Notes

When we have a good cross-validation strategy / scheme in which validation data is representative of training and real world data, we will be able to build a machine learning model which is highly generalizable.

Reference - Notes from the book Approaching (Almost) Any Machine Learning Problem by Abhishek Thakur.