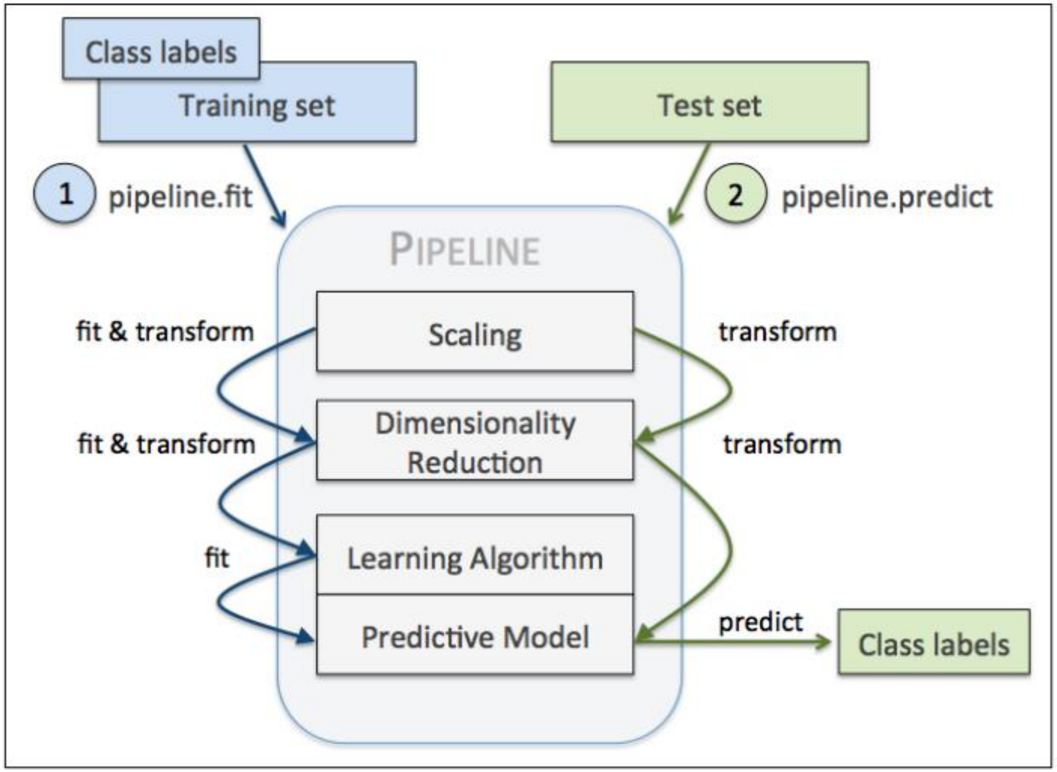
INT354
Machine Learning Foundation

Model Evaluation and Hyperparameter Tuning

Streamlining Workflows With Pipelines

- It allows us to a model including an arbitrary number of transformations steps and apply it to make predictions about new data.



Model Evaluation

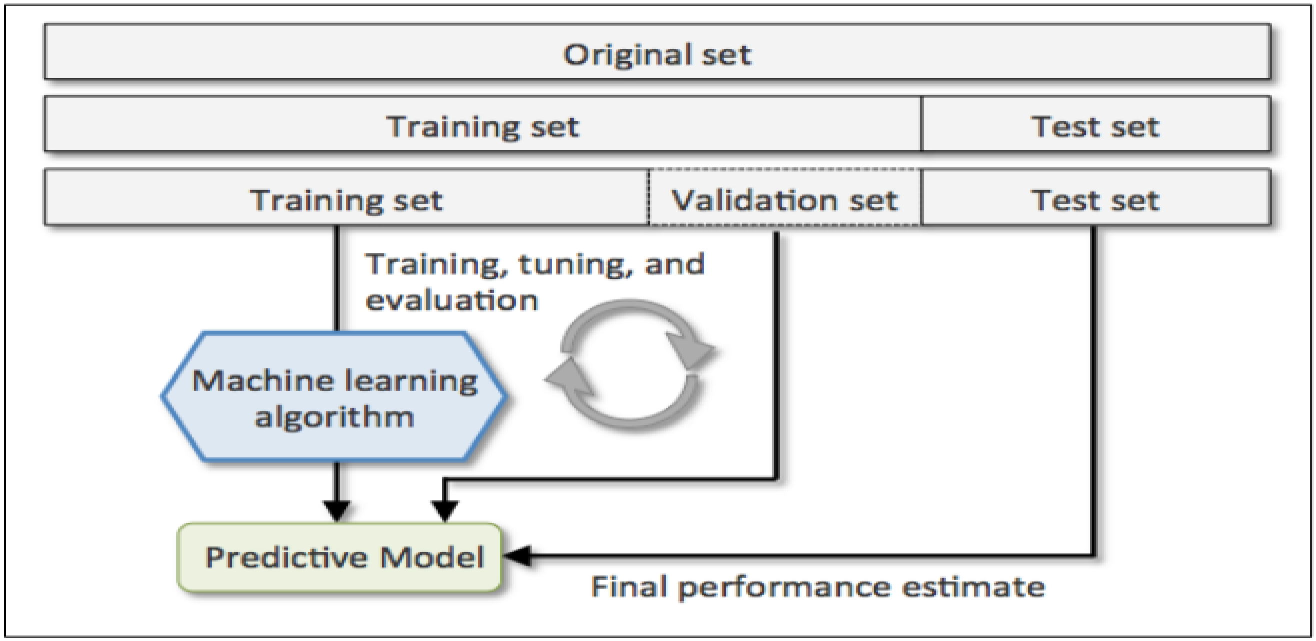
- **One of the key step in building ML model is to estimate its performance.**
- **Model can suffer from under fitting, if this is too simple (high bias).**
- **Model can suffer from over fitting, if this is too complex (high variance).**
- **To find an acceptable bias-variance tradeoff, model should be evaluated carefully.**
- **Holdout cross validation and k-fold cross validation helps us to obtain reliable estimates of the model's generalization error.**

Holdout Method

- In this, split initial dataset into separate training and test dataset – the former is used to train the model and latter is used to estimate its performance.
- We are also interested in tuning and comparing different parameters settings to further improve the performance. This process is called model selection.
- Model selection refers to a given ML problem for which we want to select the optimal values of tuning parameters, also called Hyperparameters.
- However, if we reuse the same test dataset over and over again during model selection, it will become part of training data and thus the model will be more likely to overfit.
- A better way of using the holdout method for model selection is to separate the data into three parts: a training set, a validation set and a test set.

Holdout Method

- The training set is used to train the model and the performance on the validation set is used for model selection.
- The advantage of having a test set that the model hasn't seen during the training and model selection steps is that we can obtain a less biased estimate of its ability to generalize to new data.



Holdout Method

- **A disadvantage of holdout method is that the performance estimate is sensitive to how we partition the training and validation subsets; the estimate will vary for different samples of the data.**

K-fold Cross Validation

- In K-fold cross validation, we randomly split the training data set into k folds without replacement, where k-1 folds are used for model training and one fold is used for testing.
- This procedure is repeated k times so that we obtain k models and performance estimates.
- Since K-fond cross validation is a resampling technique without replacement, the advantage of this approach is that each sample point will be part of a training and test data exactly once.
- Thus, it yields a lower variance estimate of the model performance than the holdout method.

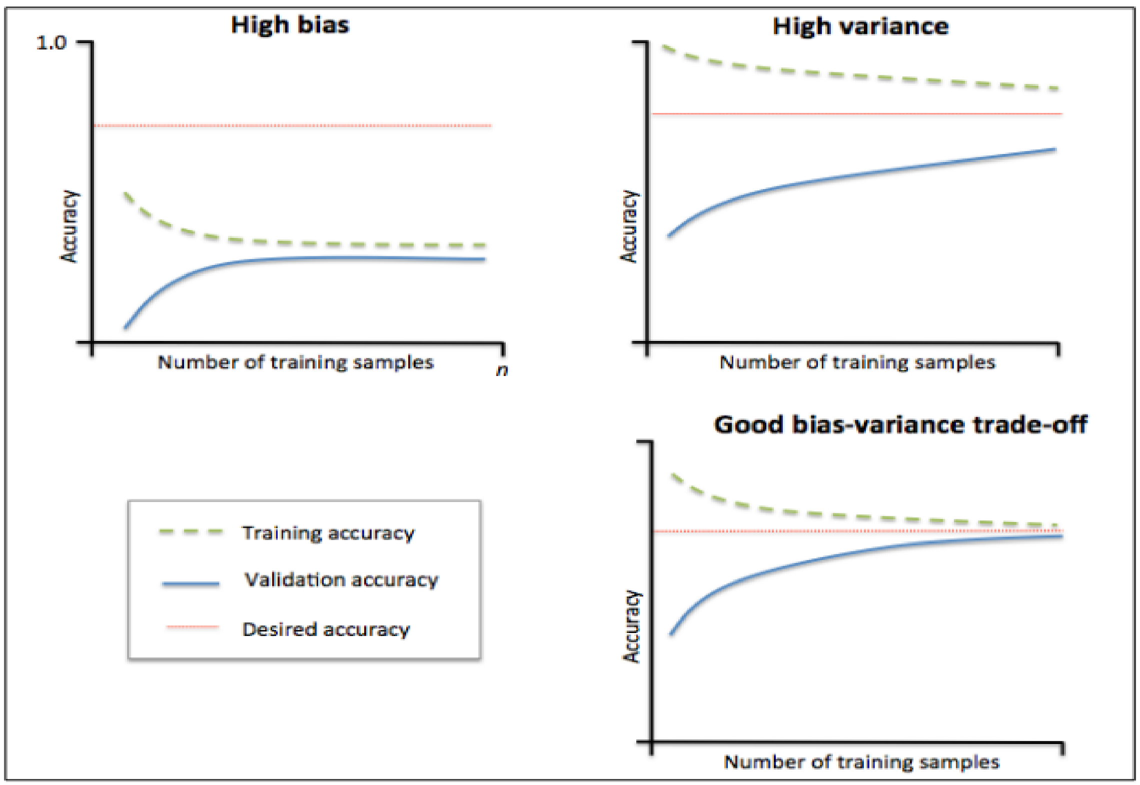
K-Fold Cross Validation

K-Fold Cross Validation

- A special case of k-fold cross validation is the leave one out (LOO) cross validation method.
- In LOO, we set the number of folds equal to the number of training samples so that only one training sample is used for testing during each iteration.
- This is recommended approach for working with small datasets.
- A slight improvement over the standard K-fold cross validation approach is stratified K-fold cross validation, which can yield better bias and variance estimates, especially in cases of unequal proportions.
- In stratified cross validation, the class proportions are preserved in each fold to ensure that each fold is representative of the class proportions in the training dataset.

Debugging Algorithms with Learning and Validation Curves

- There are basically two diagnostic tools that improve the performance of a learning algorithm: Learning curves and Validation Curves.



Tuning Hyperparameter Via Grid Search

- **It is a brute force exhaustive search paradigm where we specify a list of values for different hyperparameters, and the computer evaluates the model performance for each combination of those to obtain the optimal set.**

Confusion Matrix

- A matrix that lays out the performance of a learning algorithm.
- It reports the counts of the true positive, true negative, false positive and false negative.

		Predicted class	
		P	*N*
Actual Class	*P*	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Thanks