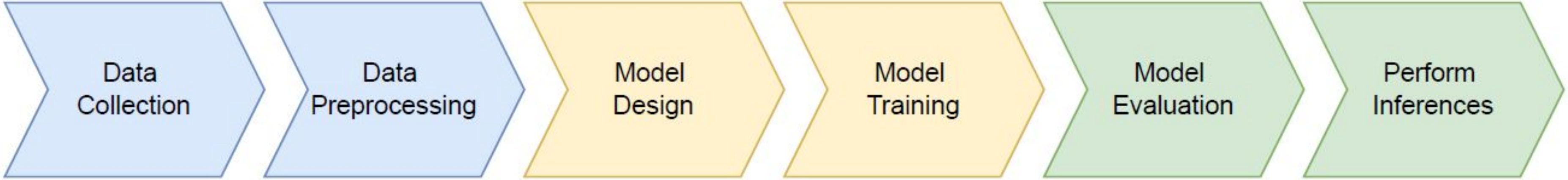
Model Evaluation

Jose Luis Paniagua Jaramillo
jlpaniagua@uao.edu.co

Machine Learning Workflow

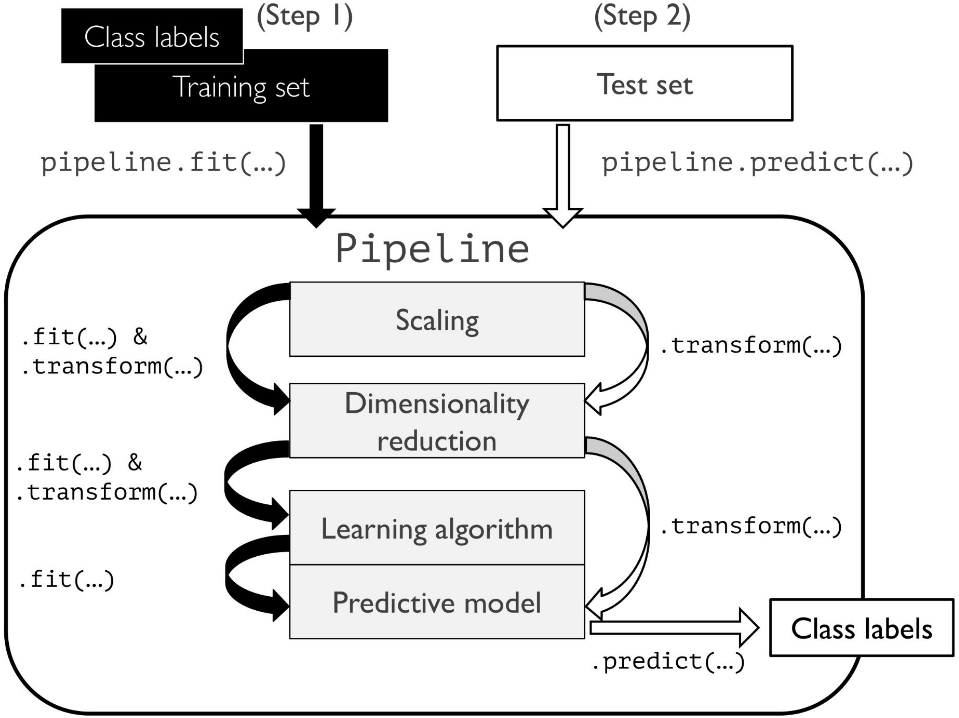


Example:

Apply the ML workflow to **Breast Cancer Wisconsin dataset**

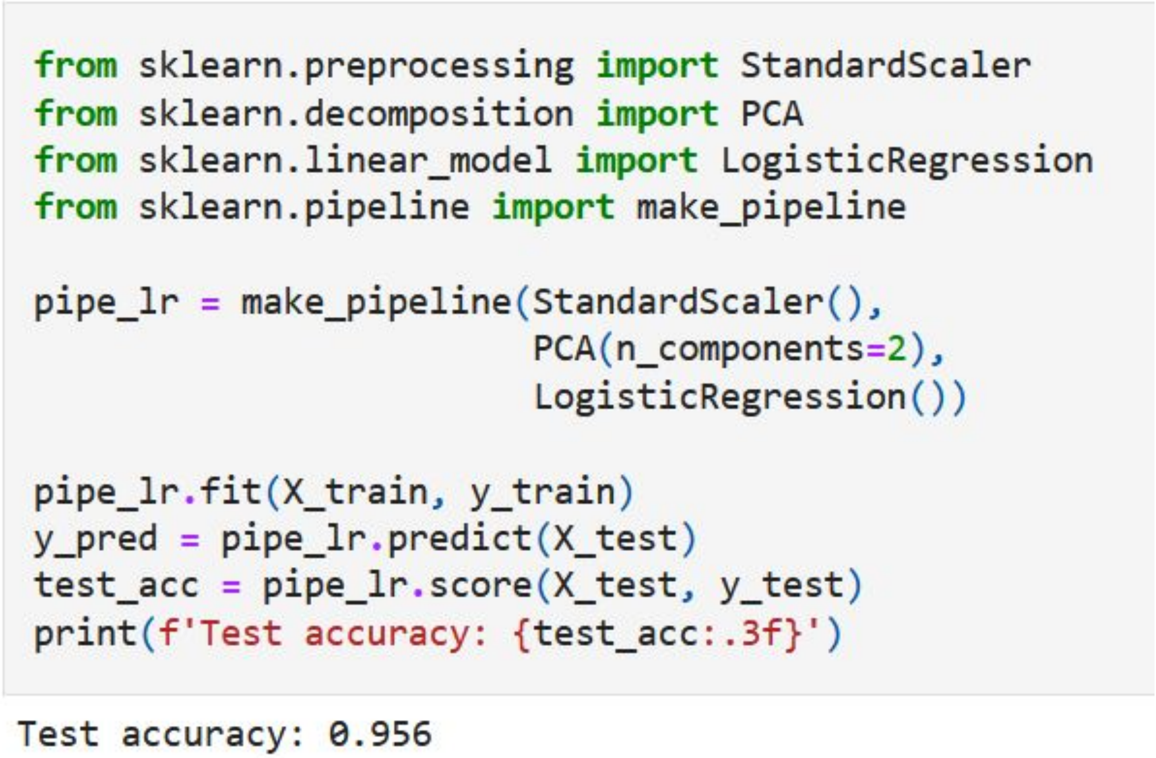
1. Use LabelEncoder to encode the class into integers
2. Use PCA as a feature extraction technique for dimensionality reduction. assume that we want to compress our data from the initial 30 dimensions into a lower two-dimensional subspace.

Pipelines



Pipelines allows us to fit a model including an arbitrary number of transformation steps and apply it to make predictions about new data.

Pipelines



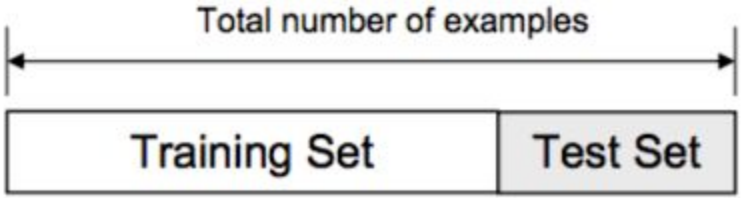
```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline

pipe_lr = make_pipeline(StandardScaler(),
                        PCA(n_components=2),
                        LogisticRegression())

pipe_lr.fit(X_train, y_train)
y_pred = pipe_lr.predict(X_test)
test_acc = pipe_lr.score(X_test, y_test)
print(f'Test accuracy: {test_acc:.3f}')
```

```
Test accuracy: 0.956
```

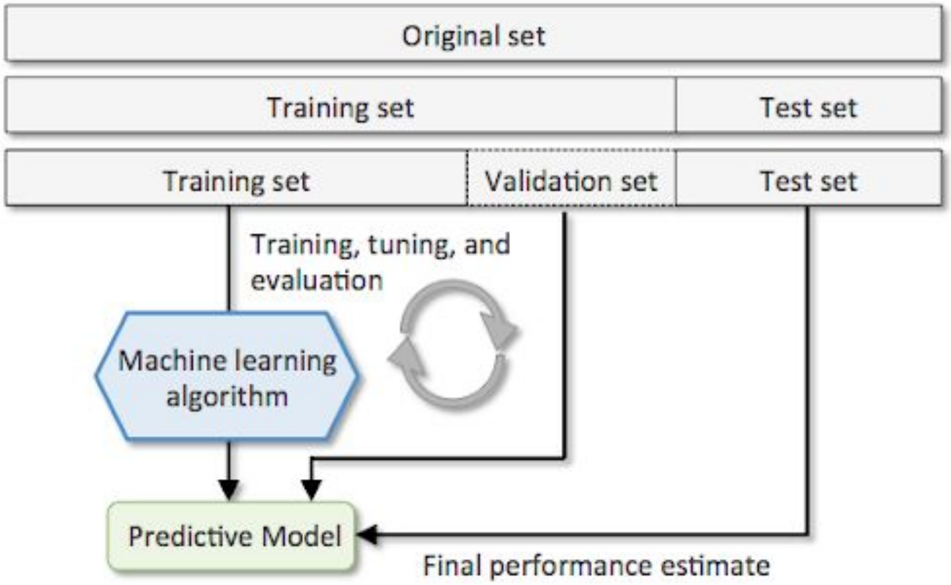
Train-Test-Split Validation (the holdout method)



- In typical machine learning applications, we are also interested in tuning and comparing different parameter settings to further improve the performance for making predictions on unseen data. (***model selection and hyperparameter tuning***)

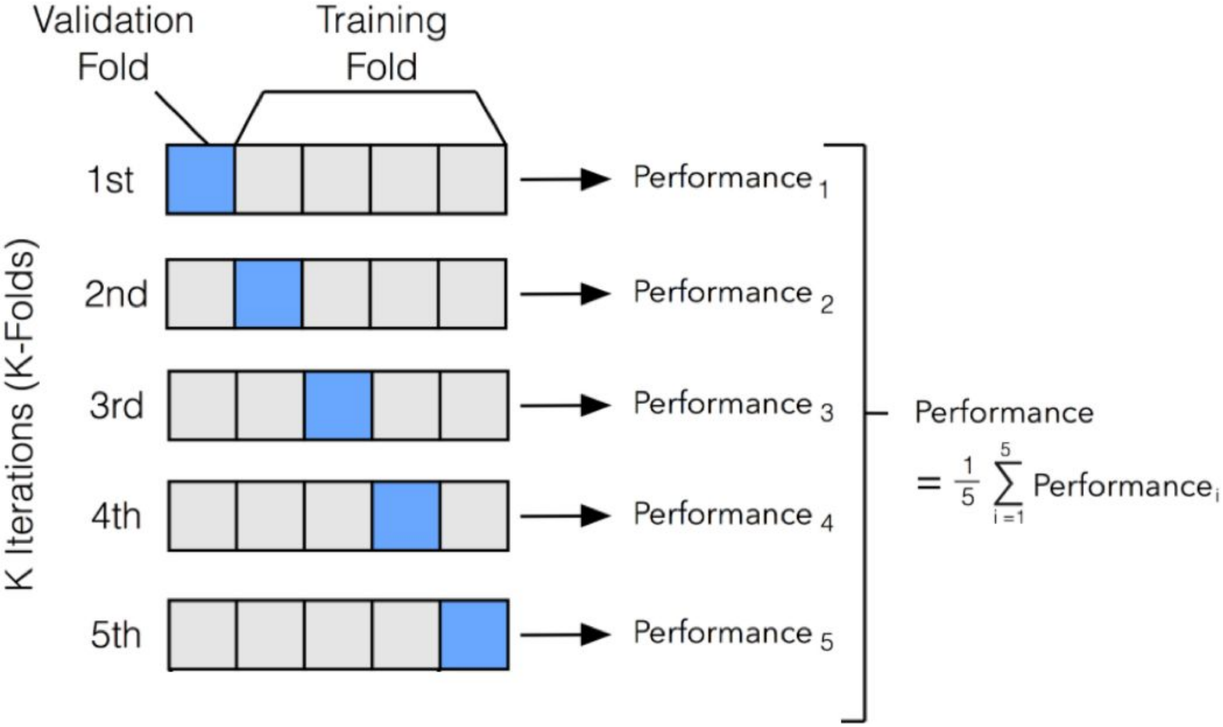
- if we reuse the same test dataset over and over again during model selection, it will become part of our training data and thus the model will be more likely to **overfit.**

Holdout cross-validation


Original set
Training set
Test set
Training set
Validation set
Test set
Training, tuning, and evaluation
Machine learning algorithm
Predictive Model
Final performance estimate

- A disadvantage of the holdout method is that the performance estimate may be very sensitive to how we partition the training dataset into the training and validation subsets.

K-fold Cross-Validation

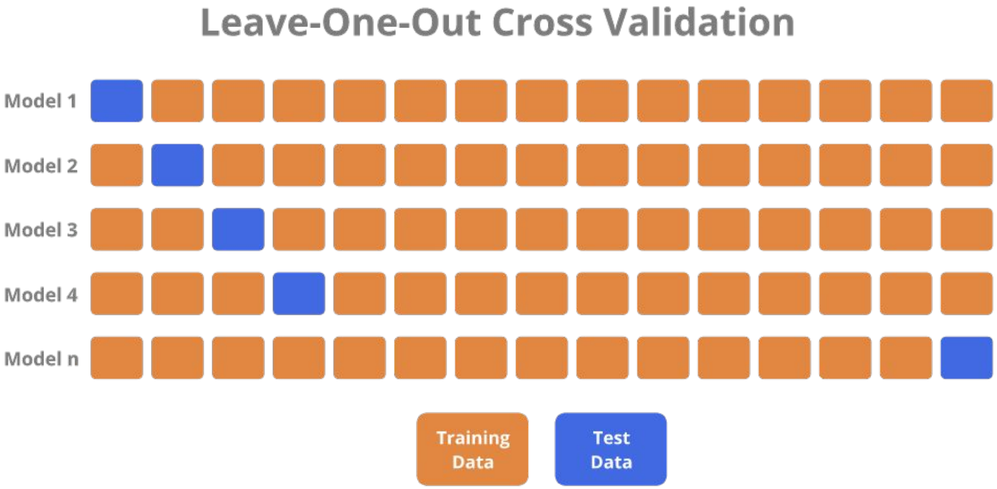


- the advantage of this approach is that in each iteration, each example will be used exactly once.

- in k-fold cross-validation all data points are being used for evaluation.

Leave-one-out cross-validation

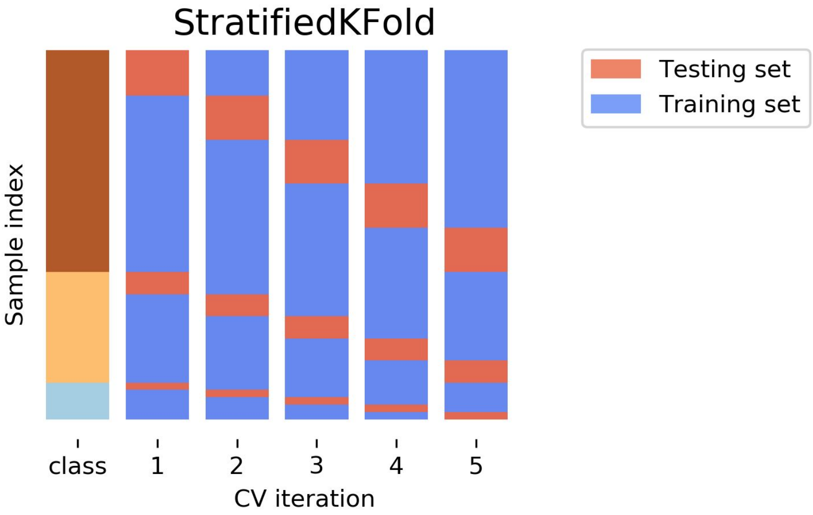
we set the number of folds equal to the number of training examples (k = n) so that only one training example is used for testing during each iteration, which is a recommended approach for working with very small datasets.



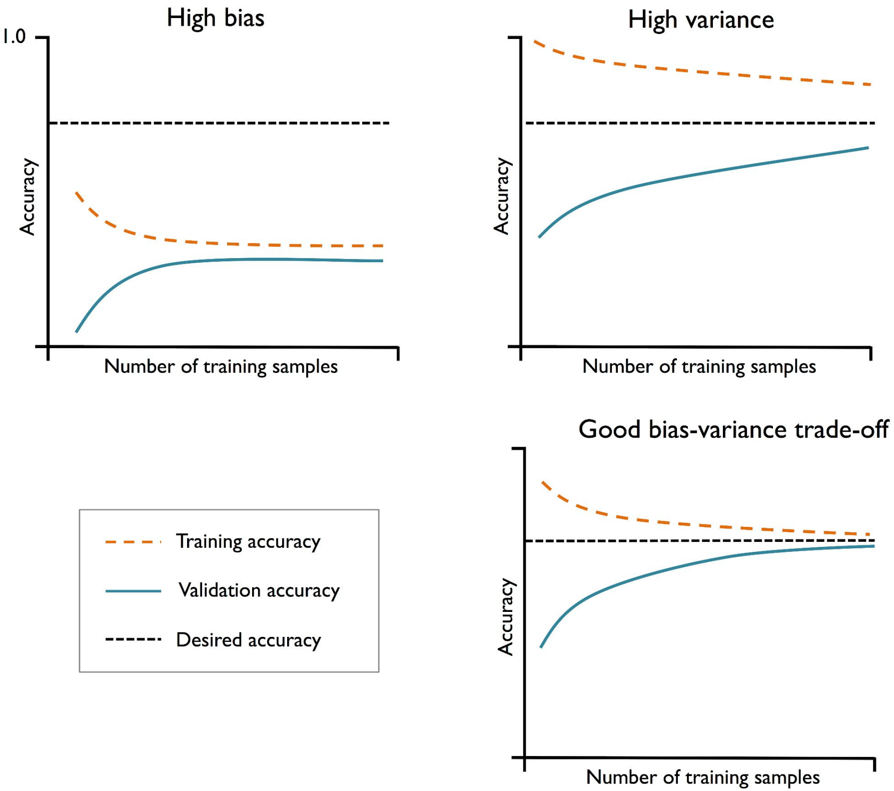
Stratified k-fold cross-validation

- The class label proportions are preserved in each fold to ensure that each fold is representative of the class proportions in the training dataset.

- Is very useful in cases of unequal class proportions.

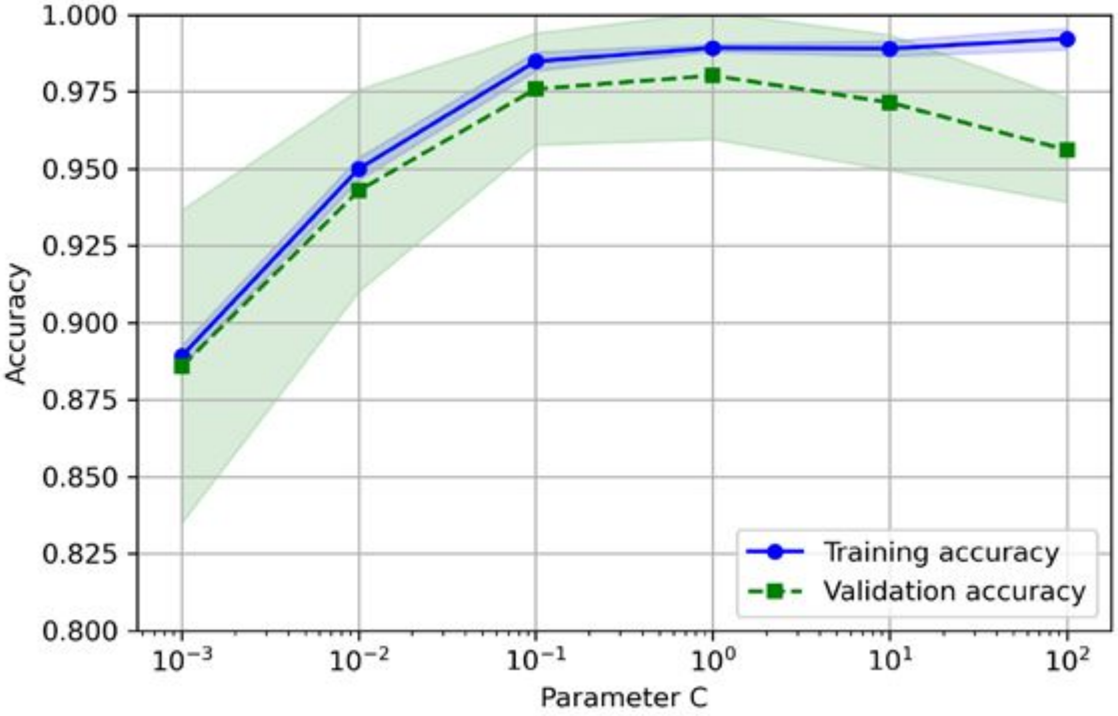


Learning Curves



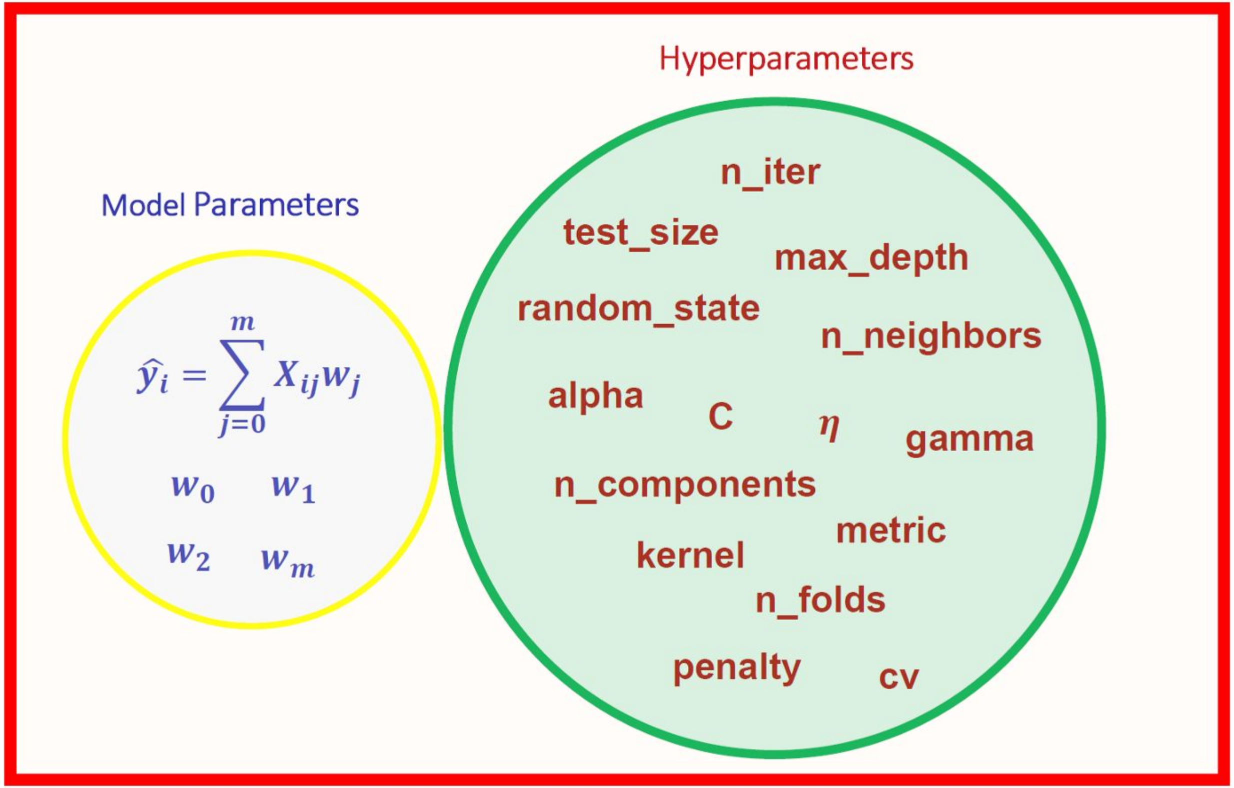
- We can use learning curves to diagnose whether a learning algorithm has a problem with overfitting (high variance) or underfitting (high bias).

Validation Curves

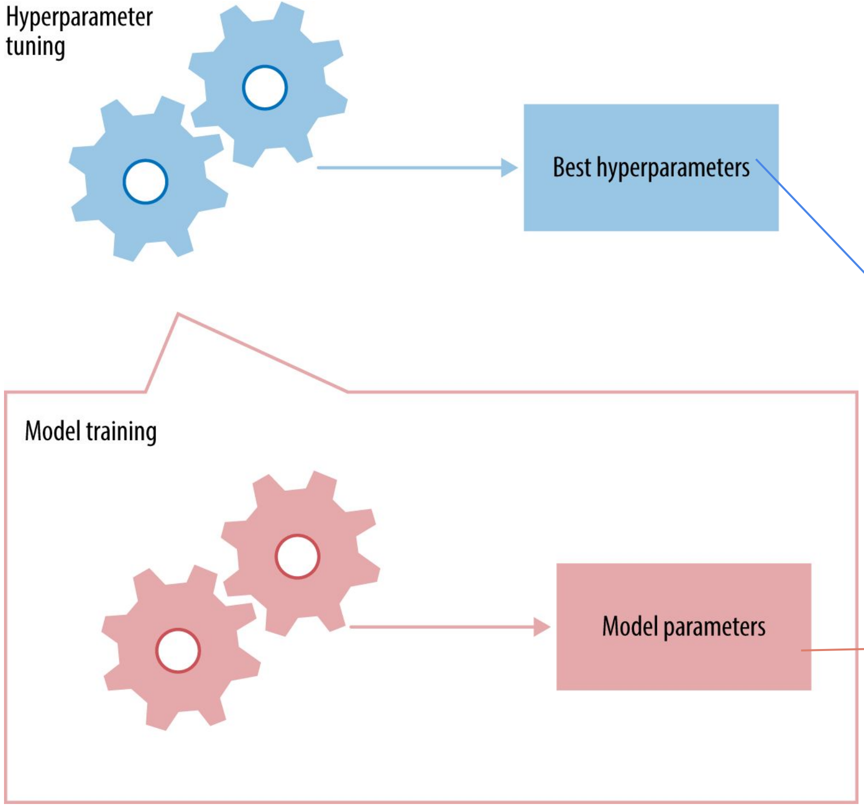


- Validation curves are a useful tool for improving the performance of a model.
- It Allow us to know the hyperparameter influence in the accuracy of the model.

Tuning Hyperparameters



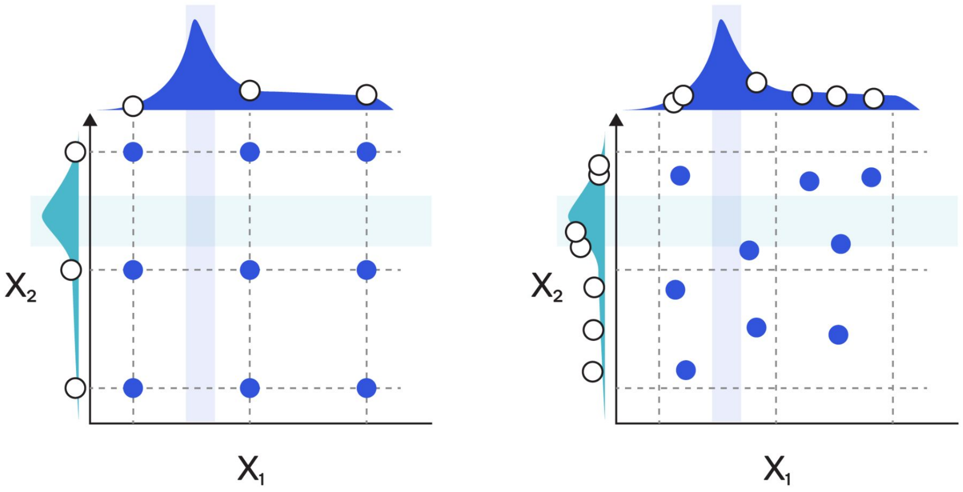
Tuning Hyperparameters



we have two types of parameters:

- The parameters of a learning algorithm that are optimized separately.
- Those that are learned from the training data, for example, the weights in logistic regression.

Grid Search



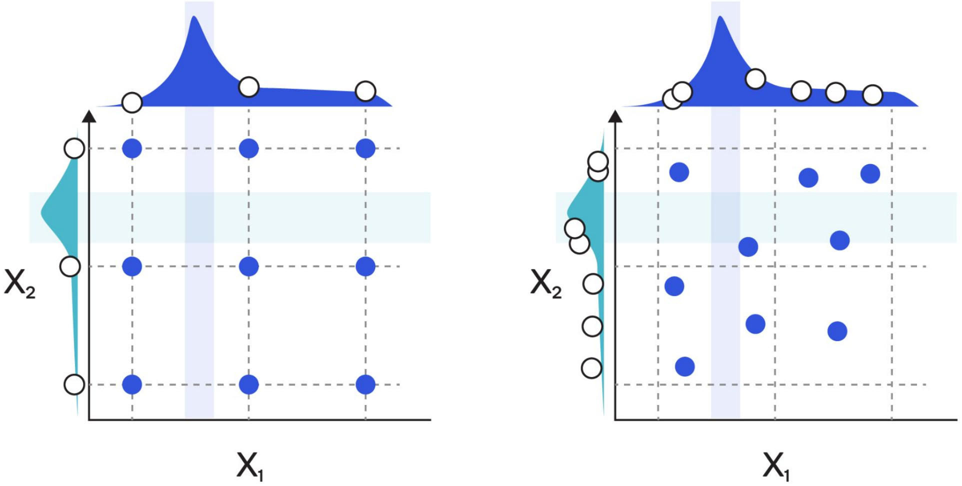
(a) Standard Grid Search

(b) Random Search

- GS can help to improve the performance of a model by finding the optimal combination of hyperparameter values.

- It’s a brute-force exhaustive search paradigm where we specify a list of values for different hyperparameters

Grid Search



(a) Standard Grid Search

(b) Random Search

- GS can help to improve the performance of a model by finding the optimal combination of hyperparameter values.

- It’s a brute-force exhaustive search paradigm where we specify a list of values for different hyperparameters

Metrics for Model Evaluation

- Choosing the right metric is crucial to evaluate ML models.

- Various metrics are proposed to evaluate ML models in different applications.

- Looking at a single metric may not always give you the whole picture of the problem you are solving, so you may need to use two or more and compare them.

Metric vs Loss Function

- Metric is different from loss function.

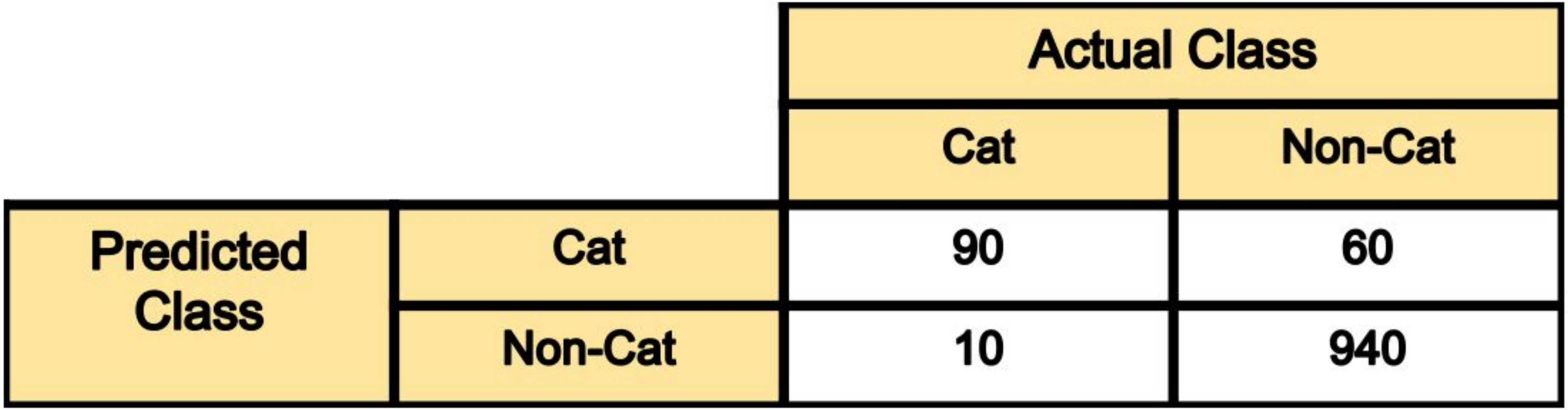
- Loss functions are used to train a model, using some kind of optimization, and are usually differentiable in model’s parameters.

- Metrics are used to monitor and measure the performance of a model during training, and test, and do not need to be differentiable.

- However some metrics can be used both as a loss function and a metric, such as MSE.

Classification Metrics

Confusion Matrix



		Actual Class	
		Cat	Non-Cat
Predicted Class	Cat	90	60
	Non-Cat	10	940

Classification Metrics

Classification Accuracy

Number of correct predictions divided by the total number of predictions,multiplied by 100

- Cat Example:
    - Classification accuracy= (90+940)/(1000+100)= 1030/1100= 93.6%

Classification Metrics

Precision

- In many cases, accuracy is not a good indicator. One of these scenarios is when your class distribution is imbalanced
    - Cat example:
        - If the model predicts all samples as non-cat, it would result in a 1000/1100= 90.9%.
        - Precision= True_Positive/ (True_Positive+ False_Positive)
        - The precision of Cat and Non-Cat class in above example can be calculated as:
        - Precision_cat= 90/(90+60) = 60%
        - Precision_NonCat= 940/950= 98.9%

Classification Metrics

Recall

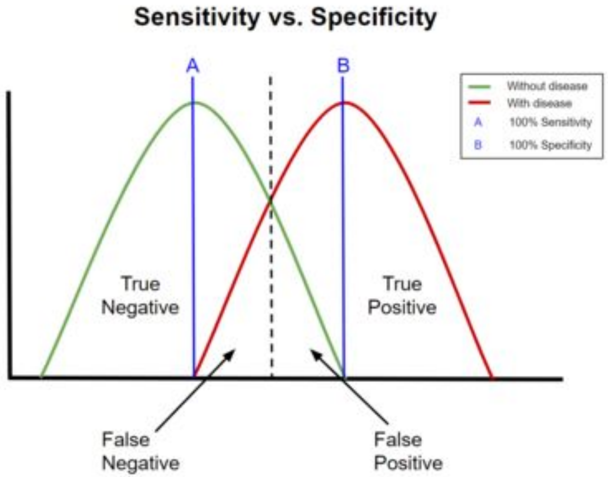
- Is defined as the fraction of samples from a class which are correctly predicted by the model.

- Recall= True_Positive/ (True_Positive+ False_Negative)
  - Cat Example:
    - Therefore, for our example above, the recall rate of cat and non-cat classes can be found as:
    - Recall_cat= 90/100= 90%
    - Recall_NonCat= 940/1000= 94%

Classification Metrics

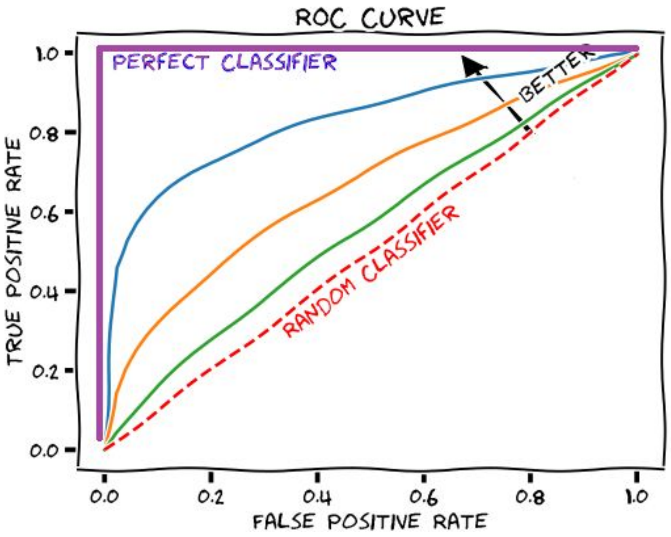
Sensitivity and Specificity

- Sensitivity= Recall= TP/(TP+FN)
- Specificity= True Negative Rate= TN/(TN+FP)



Classification Metrics

Receiver operating characteristic curve (ROC)



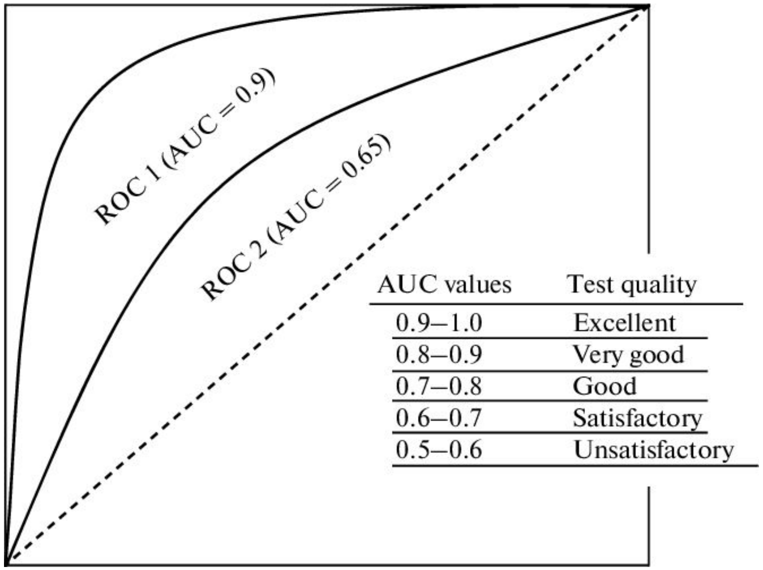
- ROC is a plot which shows the performance of a binary classifier as function of its cut-off threshold.

- The diagonal of a ROC graph can be interpreted as random guessing.

- A perfect classifier would fall into the top-left corner of the graph with a TPR of 1 and an FPR of 0.

Classification Metrics

Receiver operating characteristic curve (ROC)



AUC values	Test quality
0.9–1.0	Excellent
0.8–0.9	Very good
0.7–0.8	Good
0.6–0.7	Satisfactory
0.5–0.6	Unsatisfactory

- Based on the ROC curve, we can then compute the so-called ROC area under the curve (ROC AUC).
- In general, the higher the AUC of a model the better it is.

Regression Metrics

Mean Squared Error

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- MSE is perhaps the most popular metric used for regression problems.

- You can instead use RMSE to have a metric with scale as the target values.

Regression Metrics

Mean Absolute Error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right|$$

- MAE is another metric which finds the average absolute distance between the predicted and target values.

- MAE is known to be more robust to the outliers than MSE.

Regression Metrics

R Square

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

- R Square measures how much of variability in dependent variable can be explained by the model.
- It between 0 to 1 and bigger value indicates a better fit between prediction and actual value.
- R Square is a good measure to determine how well the model fits the dependent variables. However, it does not take into consideration of overfitting problem.

Regression Metrics

R Square adjusted

$$\text{Adjusted } R^2 = 1 - \frac{(1 - R^2)(n - 1))}{n - k - 1}$$

- Adjusted R² takes into account the features used in the predictive model.
- The more predictive features are added to the model, the higher the Adjusted R²
- The lower the Adjusted R² value, differently from what would happen with R².
- n is the number of data points and k is the number of features in the model