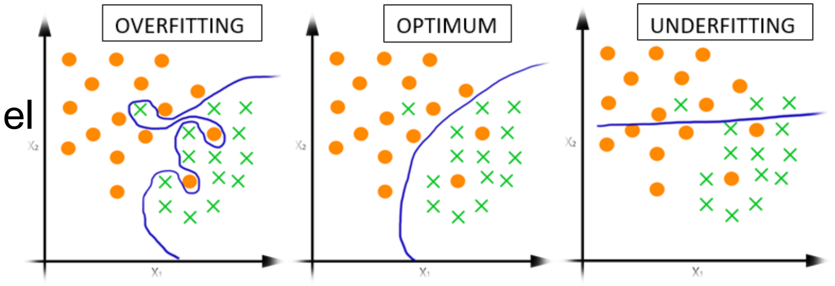
Details of ML part 4

“Evaluation is the key to making real progress in data mining, analysis, and prediction modeling”

Recall before knowing evaluation matrixes

Supervised learning: methodology

- Select model, e.g., decision tree, random forest, support vector machine, …
- Train model, i.e., determine parameters
  - Data: input + output
    - training data → determine model parameters
    - testing data → yardstick to avoid overfitting
- Prediction model
  - Data: input + output
    - validation data → final scoring of the model
- Production
  - Data: input → predict output



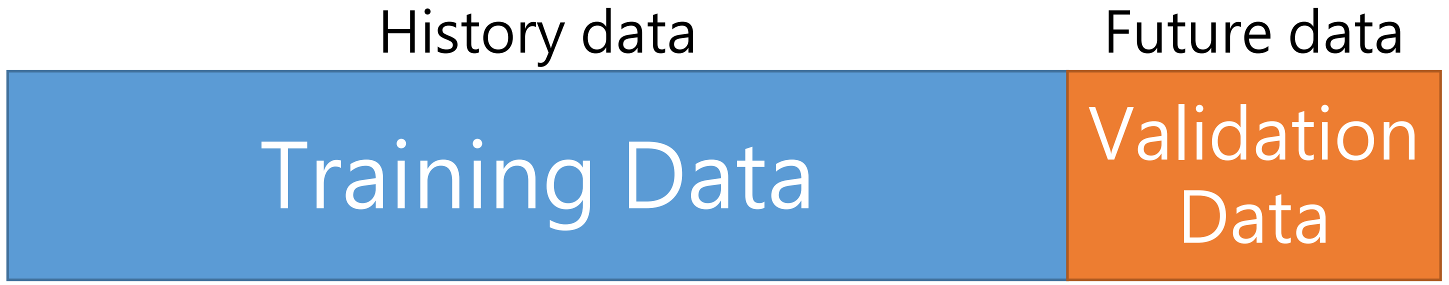
Unsupervised learning: methodology

- Select model, e.g., KMeans, DBSCAN, AgglomerativeClustering…
- Train model, i.e., select parameters
  - Data: input
    - training data
- Unsupervised learning model, e.g., clustering model
  - Data: input $\rightarrow$ result like clustering
- Production
  - Data: input $\rightarrow$ result like clustering

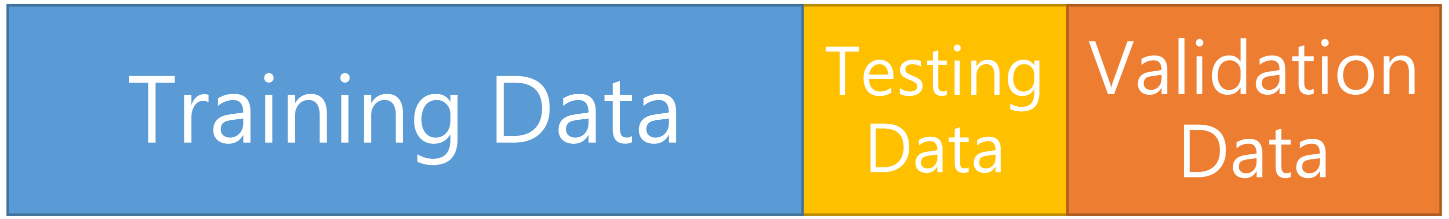
Only supervised learning needs performance measurement / evaluation

Meanings of Training, Testing, & Validation Data

- Training data → used to build (prediction) model
- Testing data → used to tune (prediction) model built by training data
- Validation data → used to validate (prediction) model, & data comes from future
- Some articles exchange words of testing & validation data, but it is fine. Most important is their meanings.



Not easy to find the best model nowadays

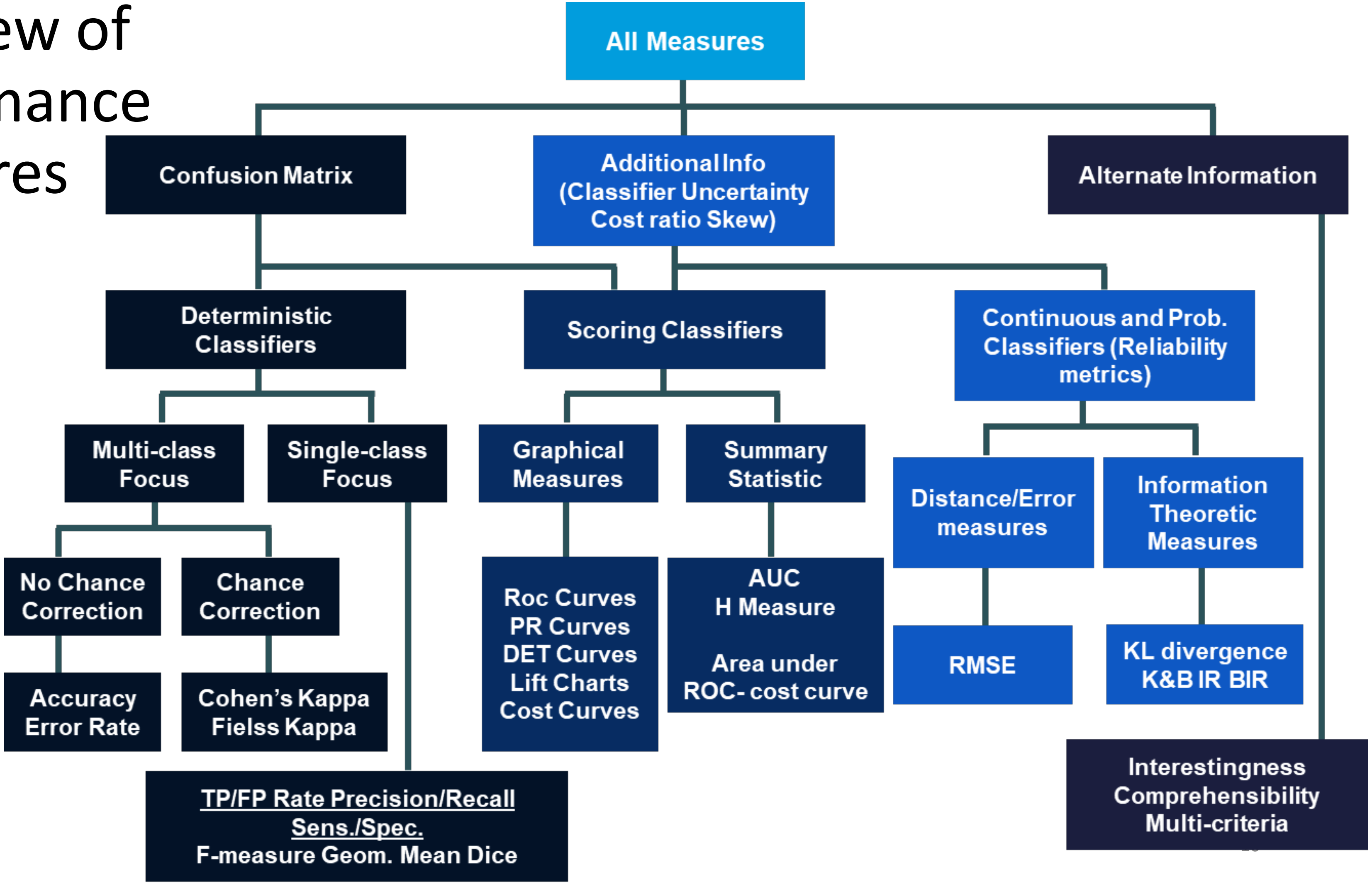


Testing data helps to find the best model

Simplest performance measurement

- Score in supervised learning estimators
  - Estimators have a score method providing a default evaluation criterion for the problem they are designed to solve.
  - Refer to each estimator's document in sklearn website
- Accuracy score in sklearn.metrics
  - Accuracy classification score
    - accuracy_score(y_true, y_pred)/accuracy_score(y_pred, y_true)

Overview of Performance Measures


All Measures
Confusion Matrix
Additional Info (Classifier Uncertainty Cost ratio Skew)
Alternate Information
Deterministic Classifiers
Scoring Classifiers
Continuous and Prob. Classifiers (Reliability metrics)
Multi-class Focus
Single-class Focus
Graphical Measures
Summary Statistic
Distance/Error measures
Information Theoretic Measures
No Chance Correction
Chance Correction
Roc Curves PR Curves DET Curves Lift Charts Cost Curves
AUC H Measure Area under ROC- cost curve
RMSE
KL divergence K&B IR BIR
Accuracy Error Rate
Cohen's Kappa Fielss Kappa
TP/FP Rate Precision/Recall Sens./Spec. F-measure Geom. Mean Dice
Interestingness Comprehensibility Multi-criteria

Confusion Matrix-Based Performance Measures

True class → ↓ Hypothesized class	Pos	Neg
Yes	TP	FP
No	FN	TN
	P=TP+FN	N=FP+TN

Confusion Matrix

- Multi-Class Focus:
  - **Accuracy** = (TP+TN)/(P+N)
- Single-Class Focus:
  - **Precision** = TP/(TP+FP)
  - **Recall** = TP/P
  - **Fallout** = FP/N
  - **Sensitivity** = TP/(TP+FN)
  - **Specificity** = TN/(FP+TN)

Aliases and other measures

- Accuracy = 1 (or 100%) - Error rate
- Recall = TPR = Hit rate = Sensitivity
- Fallout = FPR = False Alarm rate
- Precision = Positive Predictive Value (PPV)

- Negative Predictive Value (NPV) = TN/(TN+FN)
- Likelihood Ratios:
  - LR+ = Sensitivity/(1-Specificity) $\rightarrow$ higher the better
  - LR- = (1-Sensitivity)/Specificity $\rightarrow$ lower the better

Pairs of Measures and Compounded Measures

- Precision / Recall
- Sensitivity / Specificity
- Likelihood Ratios (LR+ and LR-)
- Positive / Negative Predictive Values

- F-Measure:
  - F_α = [(1+ α) (Precision x Recall)]/ [(α x Precision) + Recall]   α= 1, 2, 0.5
- G-Mean: 2-class version   single-class version
  - G-Mean =   Sqrt(TPR x TNR)   or   Sqrt(TPR x Precision)

Skew and Cost considerations

- Skew-Sensitive Assessments: e.g., Class Ratios
  - Ratio+ = (TP + FN)/(FP + TN)
  - Ratio- = (FP + TN)/(TP + FN)
  - TPR can be weighted by Ratio+ and TNR can be weighted by Ratio-

- Asymmetric Misclassification Costs:
  - If the costs are known,
    - Weigh each non-diagonal entries of the confusion matrix by the appropriate misclassification costs
    - Compute the weighted version of any previously discussed evaluation measure.

Some issues with performance measures

True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

True class →	Pos	Neg
Yes	400	300
No	100	200
	P=500	N=500

- Both classifiers obtain 60% accuracy
- They exhibit very different behaviours:
    - On the left: weak positive recognition rate/strong negative recognition rate
    - On the right: strong positive recognition rate/weak negative recognition rate

Some issues with performance measures (cont'd)

True class →	Pos	Neg
Yes	500	5
No	0	0
	P=500	N=5

True class →	Pos	Neg
Yes	450	1
No	50	4
	P=500	N=5

- The classifier on the left obtains 99.01% accuracy while the classifier on the right obtains 89.9%
    - Yet, the classifier on the right is much more sophisticated than the classifier on the left, which just labels everything as positive and misses all the negative examples.

Some issues with performance measures (cont'd)

True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

True class →	Pos	Neg
Yes	200	100
No	300	0
	P=500	N=100

- Both classifiers obtain the same precision and recall values of 66.7% and 40% (Note: the data sets are different)
- They exhibit very different behaviours:
  - Same positive recognition rate
  - Extremely different negative recognition rate: strong on the left / nil on the right
- Note: Accuracy has no problem catching this!

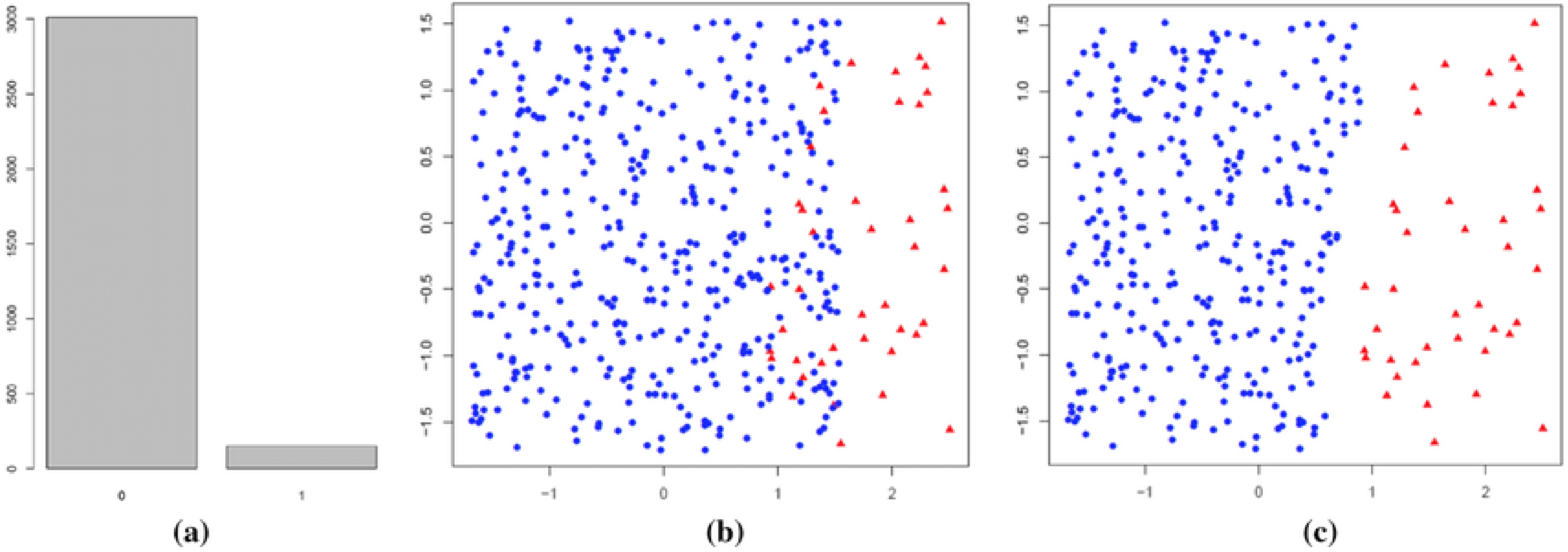
Observations

- This and similar analyses reveal that each performance measure will convey some information and hide other.
- Therefore, there is an information trade-off carried through the different metrics.
- A practitioner has to choose, quite carefully, the quantity s/he is interested in monitoring, while keeping in mind that other values matter as well.
- Classifiers may rank differently depending on the metrics along which they are being compared.

However, … (no sample / standard answers)

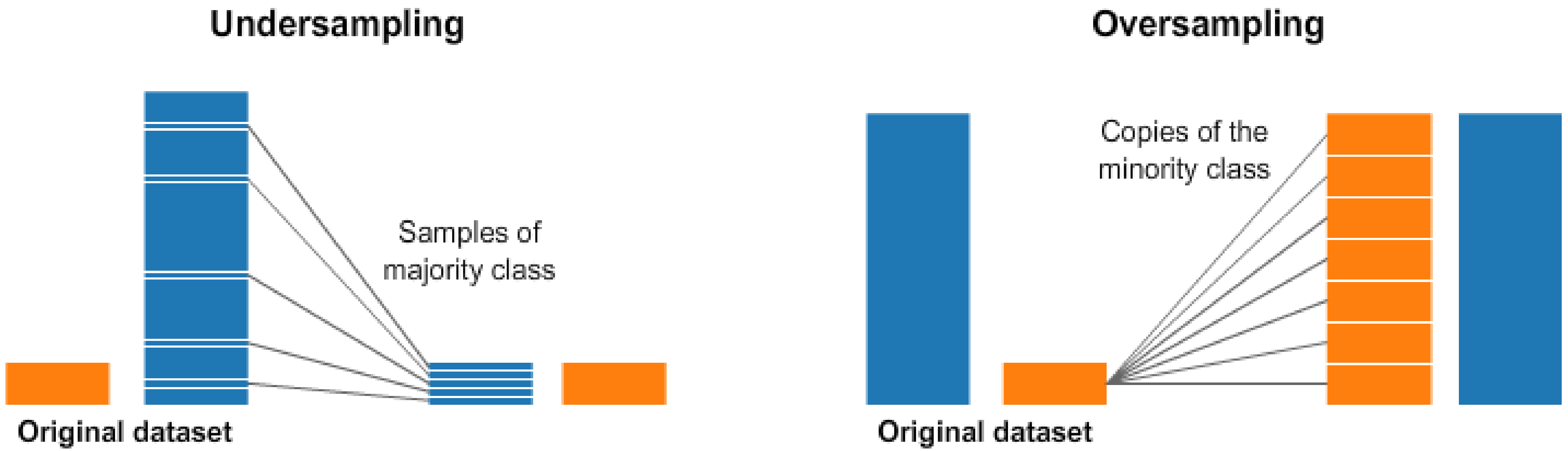
- Evaluation gives us a lot of, sometimes contradictory, information. How do we make sense of it all?
- Evaluation standards differ from person to person and discipline to discipline. How do we decide which standards are right for us?
- Evaluation gives us support for a theory over another, but rarely, if ever, certainty. So where does that leave us?

Imbalanced Dataset



https://www.researchgate.net/figure/Undersampling-imbalanced-datasets-a-Imbalanced-dataset-b-overlapping-region-and-c_fig1_343052963

Two Resamplings



Kaggle

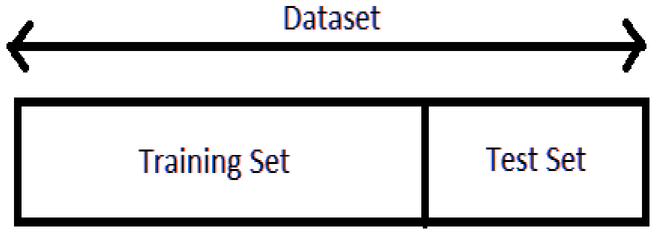
CROSS VALIDATION:

Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set.

The three steps involved in cross-validation are as follows :

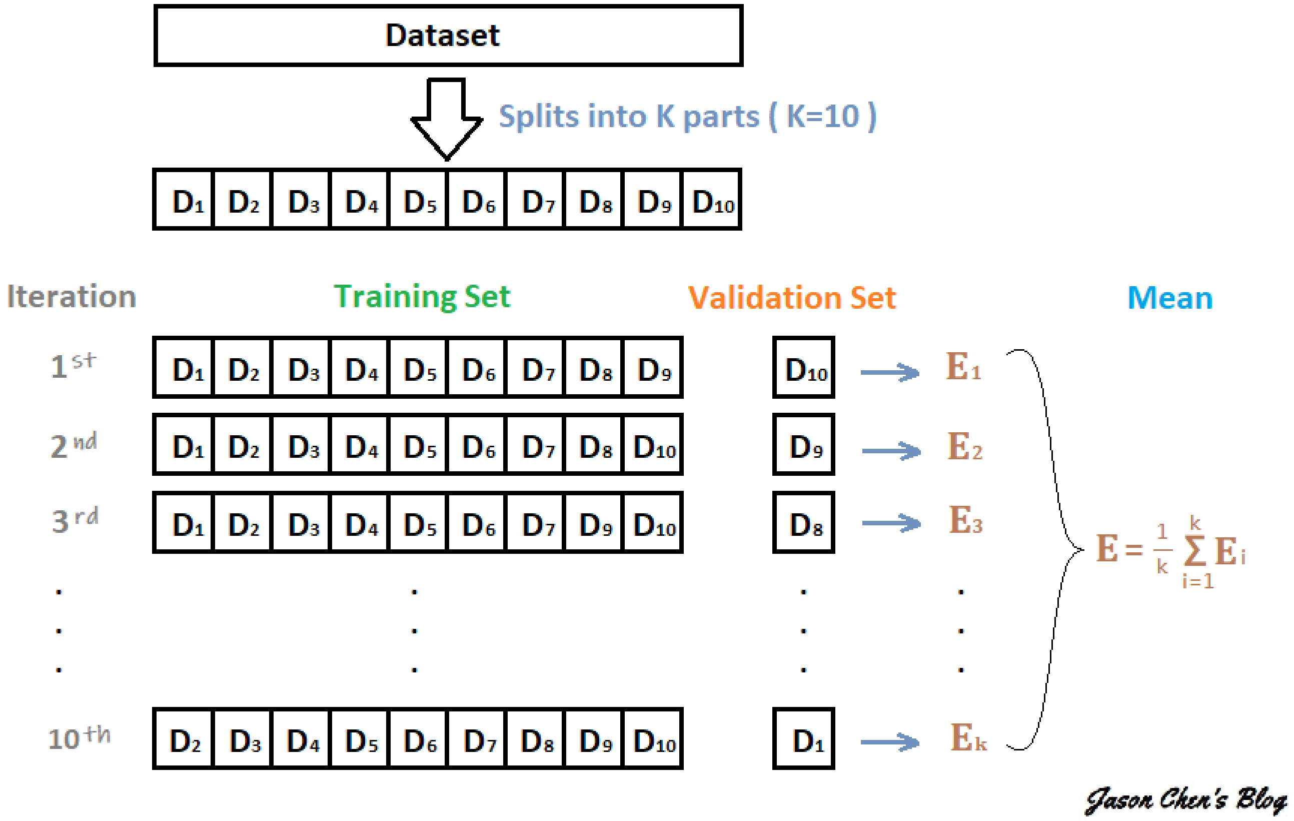
1. Split data set into training and test set
2. Using the training set train the model.
3. Test the model using the test set

USE: To get good out of sample accuracy



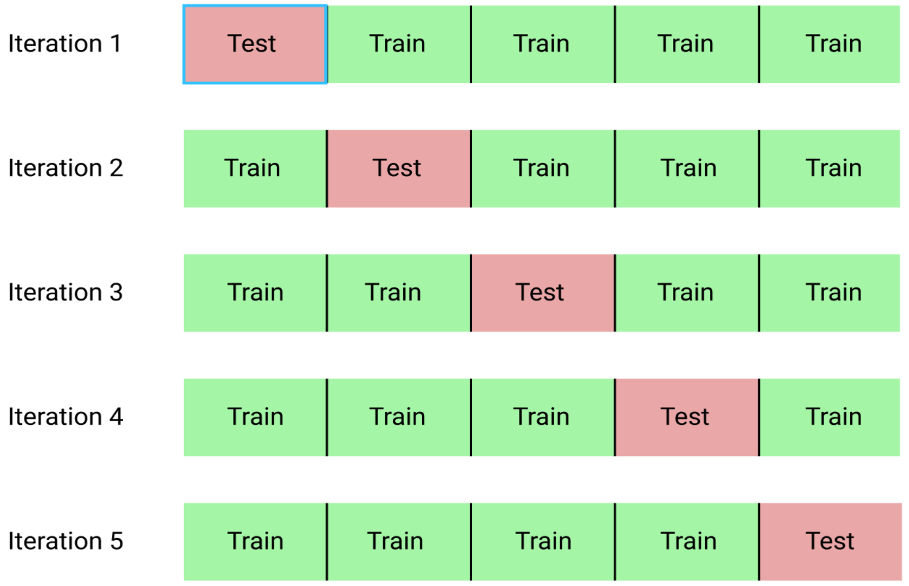
Even though we use cross validation technique we get variation in accuracy when we train our model for that we use K-fold cross validation technique

K-fold Cross-Validation


Dataset
Splits into K parts (K=10)
D_1 D_2 D_3 D_4 D_5 D_6 D_7 D_8 D_9 D_{10}
Iteration
Training Set
Validation Set
Mean
1st
D_1 D_2 D_3 D_4 D_5 D_6 D_7 D_8 D_9
D_{10}
E_1
2nd
D_1 D_2 D_3 D_4 D_5 D_6 D_7 D_8 D_{10}
D_9
E_2
3rd
D_1 D_2 D_3 D_4 D_5 D_6 D_7 D_9 D_{10}
D_8
E_3
10th
D_2 D_3 D_4 D_5 D_6 D_7 D_8 D_9 D_{10}
D_1
E_k
$E = \frac{1}{k}\sum_{i=1}^{k} E_i$
Jason Chen's Blog

https://jason-chen-1992.weebly.com/home/-cross-validation

In **K-fold cross validation**, we split the data-set 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.



Variations on K-fold Cross-Validation

- Nested K-Fold Cross Validation
  - https://scikit-learn.org/stable/auto_examples/model_selection/plot_nested_cross_validation_iris.html
- Repeated K-Fold
  - https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RepeatedKFold.html#sklearn.model_selection.RepeatedKFold
- Stratified K-Fold
  - https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html#sklearn.model_selection.StratifiedKFold
  - https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html#sklearn.model_selection.StratifiedShuffleSplit
- Group K-Fold
  - https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupKFold.html#sklearn.model_selection.GroupKFold

Tuning Model Manually

1. Select parameters
2. Set values to these parameters
3. Write for loops
   (number of for loops equals to number of selected parameters)

ML	Model	Usually Adjusted Parameters
Supervised	Linear Regression	fit_intercept
	PolynomialFeatures	degree
	Descision Tree	max_depth, min_samples_split, min_samples_leaf, class_weight
	Random Forest	n_estimators, max_depth, min_samples_split, min_samples_leaf, class_weight
	XGBoost	n_estimators (or num_boosting_rounds), learning_rate, max_depth
	SVM (kernel = "linear")	-----
	SVM (kernel = "RBF")	C, gamma, class_weight
	SVM (kernel = "poly")	C, gamma, class_weight, degree, coef()
	SVM (kernel = "sigmoid")	C, gamma, class_weight
	KNN	n_neighbors, weights, p
	Logistic Regression	C, penalty, solver
Unsupervised	Kmeans	n_clusters, init, algorithm
	DBSCAN	eps, min_samples
	Agglomerative Clustering	n_clusters, linkage, affinity(v1.2) / metric(v1.4)