Well-posed learning problems:
- Task: predicting action from speech
- Performance measure *P*: % of correct actions taken in user pilot study
- Experience *E*: examples of (speech, action) pairs

Inductive bias of ML algorithm is the principal by which it generalizes to unseen examples
- ID3 (DT learning with mutual info. as splitting criterion): **greedy search** for smallest tree that matches data with high **mutual information** attributes near the top
- Occam's Razor: prefer the simplest hypothesis that explains the data
- **Underfitting**: model has too strong an inductive bias (e.g. majority vote)
- Given hypothesis h, amount of **overfitting** is given by $\text{error}_{\text{true}}(h) - \text{error}(h, D_{\text{train}})$

To **avoid overfitting** for decision trees: *Test data should never be exposed to the machine learning model before test time. Using this data to tune the hyper-parameter might bias the model towards hyper-parameters that perform better on the test data than on generalized, unseen data.*

1. Do not grow tree beyond some maximum depth/minimum node size
2. Do not split if splitting criterion is below some threshold
3. Stop growing when the split is not statistically significant
4. Grow the entire tree, then prune

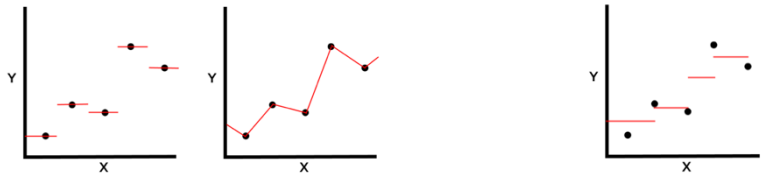
Reduced error pruning: (1) split data into two datasets, (2) grow full tree using training dataset, (3) repeatedly prune the tree:
- Replace each split with most common label among training data points that end up at node
- Evaluate each split using validation dataset: compare validation error rate with/out that split
- (Greedily) remove the split that most decreases the validation error rate
- Stop if no split improves validation

Instance-based/non-parametric/lazy learning: memorize training examples and making predictions for new instances by comparing them to the stored examples; does not construct an explicit model or generalize rules during a training phase. E.g. kNN, DT (linear, logistic (binary classification) regression, perceptrons, naïve Bayes are parametric)

1-NN (requires no training/zero training error): [Cover & Hart] Let $h(x)$ be a 1-NN binary classifier, as $N \to \infty$, $\text{error}_{\text{true}}(h) < 2 \times$ Bayes Error Rate (best possible error rate)

[k-NN] Computational efficiency: suppose N training examples with M features.

	Naive	Smart data structure: ball tree, kd-tree, neighborhood graph (exact NN)	Smart data structure (approximate NN)
Train	O(MN) or O(1)	O(N log N) if M = 1, 2; in general O(MN²) worst case	O(MN log N)
Predict	O(MN)	O(log N) // O(MN)	O(M log N)



(a) $k = 1$ (b) weighted $k = 2$ (weighted-avg of neighbors) (c) unweighted $k = 2$

Training/Model Selection: training dataset x hyperparameters → best model parameters
- Pick best model parameters by learning on training dataset for a fixed set of hyperparameters

Hyperparameter Optimization: training dataset x validation dataset → best hyperparameters
- Learning on the training data and evaluating error on the validation error
- **Never look at test errors to choose k/hyperparameters!**
- Given a finite amount of computational time, grid search > random search

Cross-Validation: training dataset x learning algorithm x loss function (e.g. 0/1 error)
- Divide data into N folds, repeatedly train on N-1 folds and predict on the held-out fold
- **Error/distance**: *Hamming distance between two vectors of {0, 1} is just the square of their Euclidean distance. While the raw magnitude of the distance metric might change, the relative order of the distance values remains the same and hence the test error doesn't change.*

Testing: test dataset x hypothesis (i.e., fixed model parameters) → test error
- Evaluate hypothesis corresponding to a decision rule with fixed model params. on test data

1. Split {train-original} into {train-subset} and {validation}. Pick the hyperparameters that when training on {train-subset} give the lowest error on {validation} [or: lowest cross-validation error on {train-original}]. Call these hyperparameters {best-hyper}.
2. Retrain a new model using {best-hyper} on {train-original} = {train-subset} ∪ {validation}.
3. Report test error by evaluating on {test}.

(Online) Perceptron: $\mathbf{x}' = (1, x_1, \dots, x_D)$, $\boldsymbol{\theta} = (b, w_1, \dots, w_D)$, then $\boldsymbol{\theta}^T\mathbf{x}' = \mathbf{w}^T\mathbf{x} + b$
- Receive $\mathbf{x}'^{(t)}$, predict $\hat{y} = \text{sign}\left(\boldsymbol{\theta}^T\mathbf{x}'^{(t)}\right)$, compare to true label $y^{(t)}$
- If misclassified, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + y^{(t)}\mathbf{x}'^{(t)}$, repeat while not converged (i.e., no mistakes)

Dataset $\mathfrak{D}$ is **linearly separable** if ∃ linear decision boundary $\boldsymbol{\theta} = (\mathbf{w}, b)$ that perfectly classifies all examples in $\mathfrak{D}$; the margin $\gamma_{\boldsymbol{\theta}}$ of any separator $\boldsymbol{\theta}$ is the distance of the closest example in $\mathfrak{D}$ to that separator; the margin γ of $\mathfrak{D}$ is the largest margin of any linear separator.
- Let $\mathbf{x}'$ be an arbitrary point on the hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ and $\mathbf{x}''$ be an arbitrary point, then the magnitude of projection of $\mathbf{x}'' - \mathbf{x}'$ onto $\frac{\mathbf{w}}{|\mathbf{w}|_2}$ is $\frac{|\mathbf{w}^T\mathbf{x}'' + b|}{|\mathbf{w}|_2}$ (if $\mathbf{x}''$ on plane, $\mathbf{w}^T(\mathbf{x}'' - \mathbf{x}') = 0$)
- **Bias term**: shifting points after projection onto weight vector

Perceptron Mistake Bound: if the examples seen by the Perceptron Learning Algorithm (online or batch) (1) lie in a ball of radius R (centered around the origin) and (2) have a margin of $\gamma > 0$, then the algorithm makes at most (R/γ)^2 mistakes total (i.e., converge in finite steps)

MSE objective function: $J_D(\mathbf{w}, b) = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - \left(\mathbf{w}^T\mathbf{x}^{(i)} + b\right)\right)^2$

Gradient Descent: $\mathbf{g} \leftarrow \nabla_\theta J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{\theta}^T\mathbf{x}^{(i)} - y^{(i)}\right)\mathbf{x}^{(i)}$, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma\mathbf{g}$

Entropy and information gain: $I(Y \mid X) = 0$ if and only if X, Y are independent
- $H(Y) = -\sum_{y \in values(Y)} P(Y = y)\log_2 P(Y = y)$
- $H(Y \mid X = x) = -\sum_{y \in values(Y)} P(Y = y|X = x)\log_2 P(Y = y|X = x)$
- $H(Y \mid X) = \sum_{x \in values(X)} P(X = x)H(Y \mid X = x)$
- $I(Y; X) = H(Y) - H(Y \mid X) = H(X) - H(X \mid Y)$