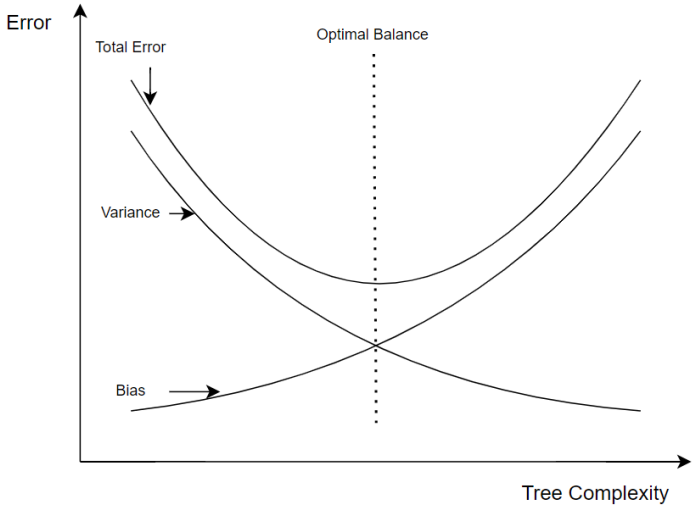
Extra Credit Assignment

Question 1: Decision Trees

A

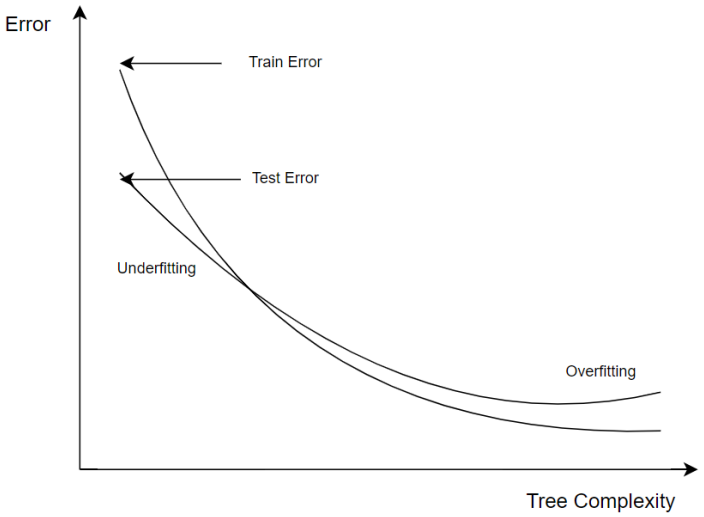
With increase in tree complexity, variance increases and bias decreases which leads to overfitting, while if tree is not complex variance is low and tree can be highly biased towards a few classes.



B

As in diagram in (A),

- Train error – bias/variance error
- Test error – total error



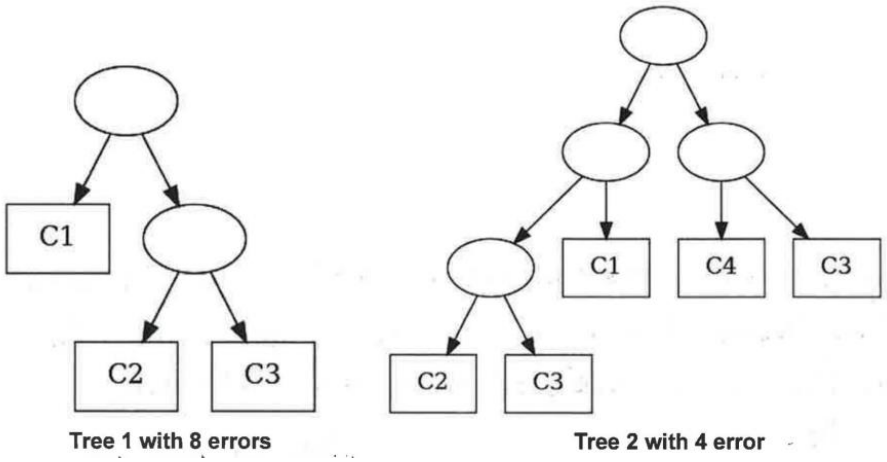
D

Purpose of tree pruning is to avoid overfitting.
Types of trees pruning techniques:

- Pre-pruning
    - Early stopping of splitting and branching at decision nodes when no significant change in error is found.

- Faster
- Post-pruning
  - Construct the entire by exhausting all the attributes.
  - Prune the trees that overfit.
  - More accurate.

E



$$Number\ of\ attributes\ =\ 16$$
$$Cost\ of\ each\ internal\ node\ in\ the\ decision\ tree\ =\ \log_2 16 = 4$$

$$Number\ of\ classes\ =\ 4$$
$$Cost\ of\ each\ leaf\ node\ =\log_2 4 = 2$$

$$Cost\ of\ each\ misclassification\ error\ =\ log_2 N$$

$$Cost(Model, Data)\ =\ Cost(Data|Model)\ +\ Cost(Model)$$

Tree 1:

$$Cost\ =\ 8\,log_2 N + (2 \times 4) + (3 \times 2) = 14 + 8\,log_2 N$$

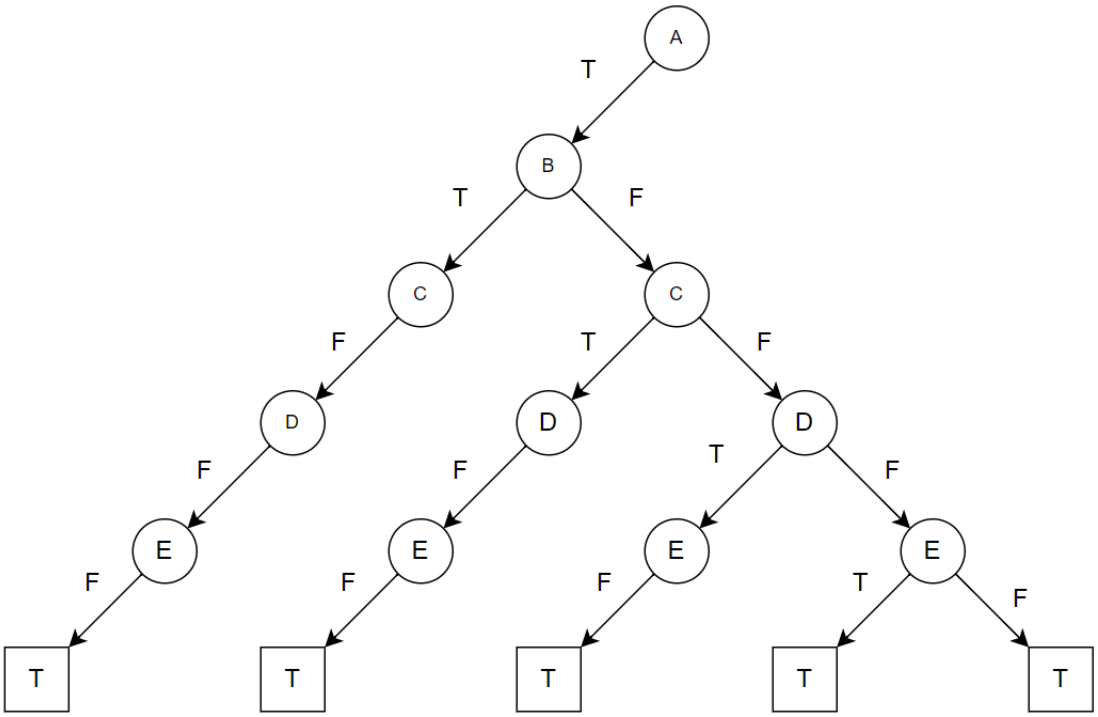
Tree 2:

$$Cost\ =\ 4\,log_2 N + (4 \times 4) + (5 \times 2) = 26 + 4\,log_2 N$$

N	Tree 1 Cost	Tree 2 Cost
2	22	30
4	30	34
8	38	38
16	46	42
32	54	46

According to the MDL principle,
If N < 16:
Tree 1 is better.
If N > 16:
Tree 2 is better.

F



Question 2: Density Estimation

A

$$l(P_g|X) = \prod_{t=1}^{N} P_g * (1 - P_g)^{x^t - 1}$$

B

$$L(P_g|X) = \log\left(\prod_{t=1}^{N} P_g * (1 - P_g)^{x^t - 1}\right) = \sum_{t=1}^{N} \log\left(P_g * (1 - P_g)^{x^t - 1}\right)$$

$$L = \sum_{t=1}^{N} \log(P_g) + \sum_{t=1}^{N} (x^t - 1)\log(1 - P_g)$$

To find the MLE of L we maximize the function by taking partial derivative of P_g and set it to 0.

$$\frac{\partial L}{\partial P_g} = 0 = \sum_t \frac{1}{P_g} - \sum_t \frac{(x^t - 1)}{1 - P_g}$$

$$\frac{N}{P_g} - \frac{1}{1 - P_g}\left(\sum_t x^t - N\right) = 0$$

$$N - NP_g = P_g \sum_t x^t - NP_g$$

$$P_g = \frac{N}{\sum_t x^t}$$

C

$$P(P_g|X) = \frac{P(X|P_g)P(P_g)}{P(X)} = \frac{P(X|P_g)P(P_g)}{\int P(X|P_g{}')P(P_g{}')dP_g}$$

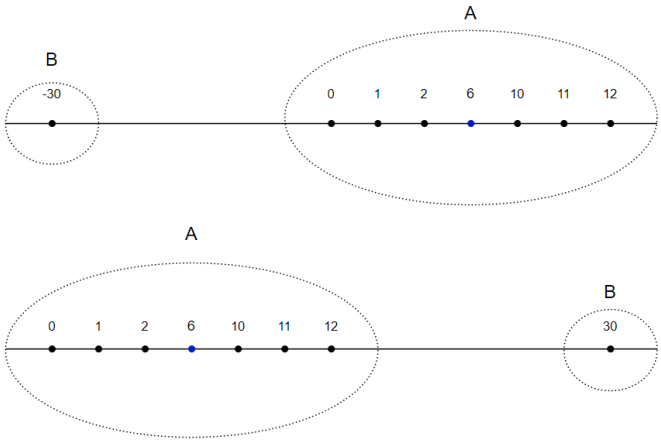
Question 3: Clustering

A

Let us consider +30 & -30 as an added point to set for 2 different cases.
For an outlier to be in its own cluster,

$$\text{dist from the kth nearest neighbor} \gg \text{average distance from the centroid of other cluster}$$

For outlier to be in its own cluster the point must be far enough from A, i.e., distance of point B from the centroid of cluster A must be large enough such that it is not in proximity to the points in A.



If the outlier is far enough from the centroids of other clusters the centroid of that point will be the point itself given the avg distance of k[th] neighbors from their respective cluster.

For cluster A,

$$A = \{0, 1, 2, 10, 11, 12\}$$
$$B = \{30\} \; or \; B = \{-30\}$$
$$Centroid \; (is \; midpoint) = \frac{0 + 12}{2} = 6$$
$$Avg \; absolute \; distance \; from \; centroid = \frac{6 + 5 + 4 + 4 + 5 + 6}{6} = 5$$
$$Distance \; from \; 30 = |6 - 30| = 24$$
$$Distance \; from \; -30 = |6 - (-30)| = 36$$

$$d_A(-30) > d_A(30) \gg avg_A(\{A\})$$

Therefore, B is an outlier, and k-means will always form a separate cluster for that point. This shows that K-means clustering is not robust with outliers.

Question 4: Dimension Reduction

A

λ_i represents the variance captured by the i[th] component.

C

- ISOMAP and Laplacian Eigenmaps are manifold learning algorithms used to discover low-dimensional embeddings of high-dimensional data.
- ISOMAP uses geodesic distances to measure the pairwise distances between data points by computing the shortest path along the manifold that connects two points.
- Laplacian Eigenmaps uses diffusion or similarity distances to measure pairwise distances by measuring the similarity between probability distributions of random walks starting from each data point.
- Both algorithms construct a weighted graph that connects nearby points, use spectral methods to compute low-dimensional embeddings of the data, and preserve local structure in the data.

Question 5: Naïve Bayes Classifier

A

Words	SPAM
Interest Free Card	No
Cash Credit Gift	Yes
Mortgage Interest Deal	No
Cash Back Credit Card	No
Debt Free Deal	No
Credit Card Interest	No
Exclusive Free Deal	Yes
Card Interest Mortgage	Yes

Using bag of words -> counting the frequency of each word in the set

$$P(Spam \mid words) = P(word_1 \mid Spam) \cdots P(word_n \mid Spam)P(Spam)$$

$$P(Spam) = \frac{3}{8}$$

$$P(Not\ Spam) = \frac{5}{8}$$

$$P(Spam \mid Credit\ Card\ Deal) = P(\ Credit \mid Spam)P(\ Card \mid Spam)P(\ Deal \mid Spam)P(Spam)$$

$$P(Spam \mid Credit\ Card\ Deal) = \frac{1}{3} * \frac{1}{3} * \frac{1}{3} * \frac{3}{8} = \frac{1}{72} = 0.014$$

$$P(Not\ Spam \mid Credit\ Card\ Deal)$$
$$= P(\ Credit \mid Not\ Spam)\ P(\ Card \mid Not\ Spam)\ \ P(\ Deal \mid Not\ Spam)\ P(Not\ Spam)$$

$$P(Not\ Spam \mid Credit\ Card\ Deal) = \frac{2}{5} * \frac{3}{5} * \frac{2}{5} * \frac{5}{8} = \frac{3}{50} = 0.06$$

$$P(Credit\ Card\ Deal \mid No) = \frac{0.06}{0.06 + 0.014} = 0.81$$

$$P(Credit\ Card\ Deal \mid Yes) = \frac{0.014}{0.06 + 0.014} = 0.19$$

$$\because P(Credit\ Card\ Deal \mid No) > P(Credit\ Card\ Deal \mid Yes)$$

Therefore, "Credit Card Deal" is NOT SPAM

B

Promotion is not present in the dataset so considering μ as Laplacian estimator where $\mu = 1$

$$P(Spam \mid Credit\ Card\ Promotion)$$
$$= P(\ Credit \mid Spam)P(\ Card \mid Spam)P(\ Promotion \mid Spam)P(Spam)$$

$$P(Spam \mid Credit\ Card\ Promotion) = \frac{1+\mu}{3} * \frac{1+\mu}{3} * \frac{0+\mu}{3} * \frac{3}{8} = 0.056$$

$$P(Not\ Spam \mid Credit\ Card\ Promotion) = \frac{2+\mu}{5} * \frac{3+\mu}{5} * \frac{2+\mu}{5} * \frac{5}{8} = 0.3$$

$$P(Credit\ Card\ Promotion \mid No) = \frac{0.3}{0.3 + 0.056} = 0.84$$

$$P(Credit\ Card\ Promotion \mid Yes) = \frac{0.056}{0.3 + 0.056} = 0.16$$

$$\because P(Credit\ Card\ Promotion \mid No) > P(Credit\ Card\ Promotion \mid Yes)$$

Therefore, "Credit Card Promotion" is NOT SPAM

Question 6: Association Rules

Words	SPAM
Interest Free Card	No
Cash Credit Gift	Yes
Mortgage Interest Deal	No
Cash Back Credit Card	No
Debt Free Deal	No
Credit Card Interest	No
Exclusive Free Deal	Yes
Card Interest Mortgage	Yes

B

$$Confidence\ (Interest \rightarrow Card) = \frac{\#\ instances\ with\ interest\ and\ card}{\#\ instances\ with\ interest}$$

$$Confidence\ (Interest \rightarrow Card) = \frac{3}{4}$$

C

$$Confidence\ (Credit\ Interest \rightarrow Card) = \frac{\#\ instances\ with\ credit,interest,card}{\#\ instances\ with\ credit,interest} \quad \text{... (a)}$$

$$Confidence\ (Credit\ Interest \rightarrow Card) = \frac{1}{1} = 1$$

This confidence is high, but the words "Credit Interest" is not frequent.

$$Confidence\ (Interest \rightarrow Card\ Credit) = \frac{\#\ instances\ with\ interest,card,credit}{\#\ instances\ with\ interest} \quad \text{... (b)}$$

$$Confidence\ (Interest \rightarrow Card\ Credit) = \frac{1}{4}$$

$$P(Interest) = \frac{4}{8} > P(Credit\ Interest) = \frac{1}{8}$$

Given the superset {card, credit, interest} having low support and its subsets (credit interest -> card) & (interest -> card credit) will have low confidence. Hence, this rule can be pruned.