**3&4**

**Classification**

Contents

[**True-False Questions and Short Answer** 2](#_Toc65770190)

[**Decision Tree Classifier** 5](#_Toc65770191)

[**Model Evaluation and Selection** 29](#_Toc65770192)

[**Rule-Based Classifier** 39](#_Toc65770193)

[**Nearest Neighbor Classifier** 45](#_Toc65770194)

[**Naïve Bayes Classifier** 46](#_Toc65770195)

[**Artificial Neural Networks** 66](#_Toc65770196)

[**Deep Learning** 67](#_Toc65770197)

[**Support Vector Machine** 68](#_Toc65770198)

[**Ensemble Methods** 70](#_Toc65770199)

[**Class Imbalance Problem, Classification Evaluation Measures, and ROC** 72](#_Toc65770200)

[**Comparison of Classification Algorithms** 109](#_Toc65770201)

# **True-False Questions and Short Answer**

1. True/False with short explanations

State whether the following statements are true or false and provide brief explanations to support your answer.

1. Pessimistic generalization error estimate is the error that the model achieves on the training set.

False. Pessimistic generalization error is the error on the training set, plus the leaf penalty factor

1. The formula of total description length is given by the following:

Cost(Model,Data) = Cost(Data|Model) + 𝛼 × Cost(Model)

Cost(Data|Model): Number of bits needed to encode the misclassified instances Cost(Model): Number of bits needed to encode the model.

Given two competing models, the model with higher total description length is preferred.

False. Higher total description length implies a more complex model and a larger number of misclassified instances, which is not preferable.

1. The classification performance of decision trees will degrade in the presence of redundant attributes (attributes are duplicates of each other).

False. In the presence of redundant attributes, any one of the similar attributes will be chosen for splitting and the classification performance will not degrade.

1. Decision Trees and RIPPER would perform equally well on a dataset in which there are many more instances of one class than the other.

False. RIPPER is better suited for handling class imbalance as it can construct ordered rule sets where rules are learned for a minority class and the majority class becomes the default class.

Decision Trees do not consider the imbalance among the classes and consider an attribute for splitting if it decreases the impurity over both the classes.

1. The presence of noise does not result in decision tree over-fitting because decision trees are resistant to noise.

False. The presence of noise can lead to the learning of a highly complex decision tree that has improved performance over noisy instances on training data but poor generalization performance over unseen test instances, which is characteristic of the phenomena of over-fitting.

1. Since the entropy for the binary split of an attribute is worse than the entropy for the multi-way, it is always preferable to use a multi-way split whenever possible.

False. Multi-way splitting can lead to excessive fragmentation of the tree, thus leading to a significant increase in its complexity. In the extreme case, this can lead to overfitting, since the multi-way split on an attribute (say student ID) may perform well on the training set but perform very poorly on the test set.

1. Larger decision trees have higher model complexity and suffer from overfitting. Hence smaller trees should always be preferred over larger trees.

False. The goal is to construct trees with good generalization performance. In scenarios where smaller trees suffer from underfitting, larger trees can be preferred over smaller trees if they show better generalization performance (are resilient to overfitting).

1. Smaller decision trees should always be preferred over larger trees as they have better generalization performance.

False. Very small decision trees can suffer from underfitting (and thus have poor generalization performance). This is why a model selection step is used to identify a decision tree that is neither too small (underfitted model that is incapable of performing well on training as well as test data) or too large (a model that overfits to the training data but performs poorly on unseen test data).

1. If the training set is such that every combination of attribute values is present in the training data and each combination is labeled either positive or negative, it is always possible to build a model using Naïve Bayes that perfectly classifies each training instance.

False. As a counterexample, NB will not be able to perfectly classify an XOR dataset built using binary attributes.

1. Decision tree and rule-based classification automatically perform variable selection.

True. At each stage of building a tree or a rule, these algorithms select which attribute to use and don’t necessarily use all attributes in the final trained model.

1. ANN is able to handle redundant attributes.

True. ANN can handle redundant attributes by assigning attribute weights and using regularization in the objective function to be minimized when training the model.

1. Nonlinear SVM is particularly effective for categorical data compared to other techniques such as decision trees.

False. It is difficult to define a kernel on categorical attributes, while a decision tree can handle categorical data just as well as continuous data. Categorical data can be binarized and still used with SVMs but the way decision trees handle categorical data is more natural.

1. SVM and Neural Network always produce the same decision boundary for a given data set with two classes.

False. Although these two techniques could produce the same decision boundary, because of the different underlying approaches and loss functions (maximizing margin vs. minimizing mean squared prediction error) used, it is more likely that this would not happen.

1. SVMs can handle missing values better than ANNs.

False. Neither technique can handle missing values.

1. Given a dataset with Y as the target label and (A, B) as the set of features, we can use Bayes theorem for computing P(Y|A, B) using the following theorem:

P(Y|A, B) = P(A, B|Y) x P(Y) / P(A, B)

True.

1. Under the Naïve Bayes assumption, we also require that P(A, B) = P(A) x P(B)

False. NB does not require P(A, B) = P(A) x P(B), but requires conditional independence among attributes, that is P(A, B | Y) = P(A|Y) x P(B|Y).

1. In Naïve Bayes, M-estimate is used to handle the case for which the class conditional probability of an attribute value is zero. Why is it important to handle this case and how does M-estimate help here?

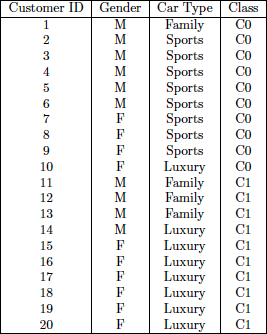
If any term goes to 0 in the calculation of the conditional probability of a specific class given the data, then that probability will also be 0. This can happen for all possible classes. In this case, there is no way to decide how to classify the object. The M-estimate always provides a non-zero estimate for the class conditional probability and thus allows the classification decision to be made based on other attributes.

1. Given a training set that contains 10,000 binary features but only 150 samples, many classification algorithms will suffer from poor performance because of the high dimensionality. One approach to address this problem is to select the 100 features that are most highly correlated with the binary class labels, and then build a model and evaluate its generalization performance using the standard k-fold cross-validation. Will this approach provide a good estimate of generalization performance?  If not, what changes would you make when building the model using cross-validation in order to improve the evaluation of the generalization performance?

#### No, if the feature selection was performed using correlation values computed from the entire data set. To ensure good generalization performance, the feature selection step must also be included as part of the cross-validation step.

# **Decision Tree Classifier**

1. Consider the training examples shown in the table below for a binary classification problem.



1. Compute the Gini index for the overall collection of training examples.

Gini = 1 − 2 x 0.25 = 0.5.

1. Compute the Gini index for the Customer ID attribute.

The Gini for each Customer ID value is 0. Therefore, the overall gini for Customer ID is 0.

1. Compute the Gini index for the Gender attribute.

The Gini for Male is 1 − 0.42 − 0.62 = 0.48. The Gini for Female is also 0.48. Therefore, the overall Gini for Gender is 0.5 x 0.48 + 0.5 x 0.48 = 0.48.

1. Compute the Gini index for the Car Type attribute using multiway split.

The Gini for Family car is 0.375, Sports car is 0, and Luxury car is 0.2188. The overall Gini is 0.1625.

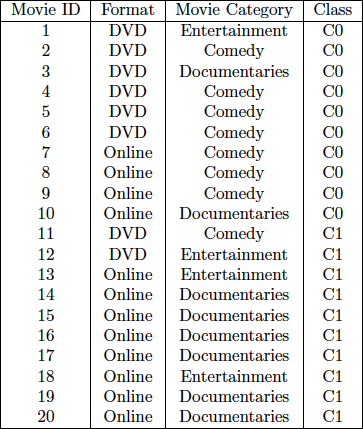
1. Which of the three attributes has the lowest Gini index?

Customer ID

1. Which of the three attributes will you use for splitting at the root node? Explain your choice.

We need to calculate GainRATIOsplit for each choice of splitting and choose the one with maximum GainRATIOsplit. The GainRATIOsplit for attributes Customer ID, Gender, and Car Type are 0.17, 0.028, and 0.39, respectively. Based on this, we choose Car Type for splitting at the root node.

1. Consider the training examples shown in the table below for a binary classification problem.



* 1. Compute the Gini, Misclassification Error, and the Entropy for the overall collection of training examples.

Gini = 0.5, Misclassification Error = 0.5, Entropy = 1.

* 1. Compute the Gini, Misclassification Error, and Entropy for all the three attributes: Movie ID, Format, and Movie Category, using multi-way splits for Movie ID and Movie Category, and binary split for Format.

Gini of Movie ID = 0, Misclassification Error of Movie ID = 0, Entropy of Movie ID = 0.

Gini of Format = 0.41, Misclassification Error of Format = 0.3, Entropy of Format = 0.875.

Gini of Movie Category = 0.3125, Misclassification Error of Movie Category = 0.2, Entropy of Movie Category = 0.399.

* 1. Compute the Information Gain for all the three attributes. Which attribute provides the highest Information Gain?

IG of Movie ID = 1, IG of Format = 0.12, IG of Movie Category = 0.601.

Movie ID provides highest IG.

* 1. Compute the Gain Ratio for all the three attributes. Which attribute provides the highest Gain Ratio?

GR of Movie ID = 1/4.32 = 0.23, GR of Format = 0.12/0.97 = 0.123, GR of Movie Category = 0.601/1.52 = 0.39. Movie Category has highest GR.

* 1. Is there a difference between the attribute that provides highest Information Gain and the attribute that provides highest Gain Ratio? Which attribute would you finally consider for splitting at the root node? Briefly explain your choice.

Movie ID provides the highest IG but Movie Category provides the highest GR. This shows that Gain Ratio can penalize the high Information Gain of Movie ID due to the large number of partitions. Hence, Movie Category should be chosen for splitting at the root node, although Movie ID has the highest Information Gain, as Movie ID clearly is just an identification attribute, and a decision tree using Movie ID would not generalize at all.

1. Consider the dataset shown in the table below for a binary classification problem.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Customer ID | Housing Type | Gender | Marital Status | Class |
|  |  |  |  |  |
| 1 | Apartment | Male | Married | C0 |
| 2 | House | Male | Single | C1 |
| 3 | House | Female | Married | C1 |
| 4 | Apartment | Female | Single | C0 |
| 5 | Apartment | Male | Married | C0 |
| 6 | Hostel | Male | Single | C1 |
| 7 | House | Female | Married | C1 |
| 8 | Apartment | Female | Single | C0 |
| 9 | Apartment | Male | Married | C0 |
| 10 | House | Male | Single | C1 |
| 11 | Hostel | Female | Married | C1 |
| 12 | House | Female | Single | C0 |
| 13 | House | Male | Married | C0 |
| 14 | Hostel | Male | Single | C1 |
| 15 | Hostel | Female | Married | C1 |
| 16 | Apartment | Female | Single | C0 |
|  |  |  |  |  |

1. Compute the Gini index, entropy, and misclassification error for the overall data.

Gini of overall data = 0**.**5

Entropy of overall data = 1

Misclassification error of overall data = 0.5

1. Compute the Gini index, entropy, and misclassification error obtained for each of the four attributes (consider a multi-way split using each unique value of an attribute).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Customer ID | Housing Type | Gender | Marital Status |
| Gini | 0 | 0.16 | 0.5 | 0.5 |
| Entropy | 0 | 0.34 | 1 | 1 |
| Misclassification Error | 0 | 0.125 | 0.5 | 0.5 |

1. Compute the Information Gain (IG) obtained by splitting the overall data using each of the four attributes. Which attribute provides the highest IG, and which attribute provides the lowest IG.

IG (Customer ID) = 1

IG (Housing Type) = 0.66

IG (Gender) = 0

IG (Marital Status) = 0

IG (Customer ID) = 1 is the highest

IG (Gender) = IG (Marital Status) = 0 is the lowest

1. Compute the Gain Ratio for splitting over each of the four attributes. Which attribute provides the highest Gain Ratio?

Gain Ratio (Customer ID) = 1/4 = 0.25

Gain Ratio (Housing Type) = 0.66/1.5 = 0.44

Gain Ratio (Gender) = 0

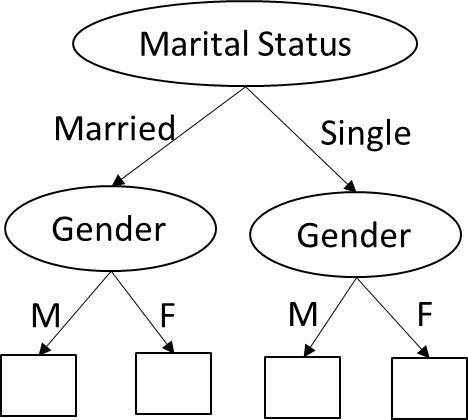
Gain Ratio (Marital Status) = 0

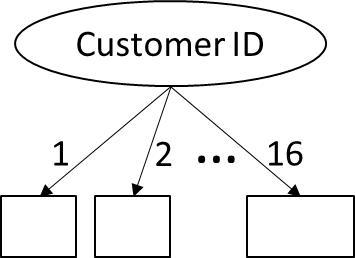
Gain Ratio (Housing Type) is highest

1. For splitting at the root node, would you choose the attribute that provides the maximum IG, or the attribute that provides maximum Gain Ratio? Briefly explain your choice.

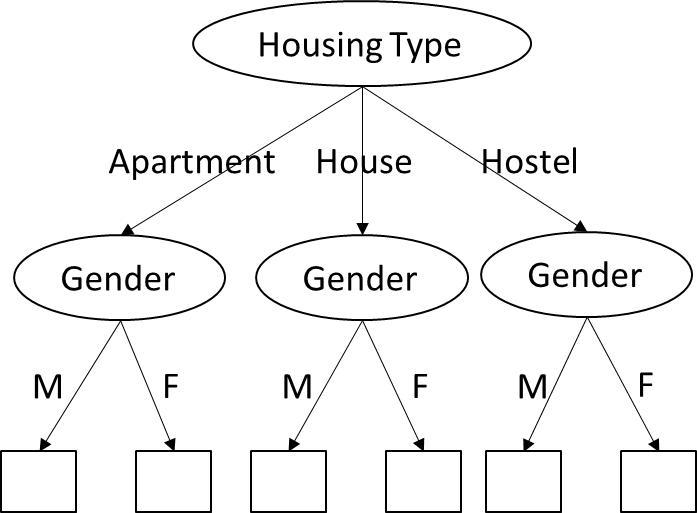
We would choose Housing Type for splitting at the root node since it provides the maximum Gain Ratio. Gain Ratio penalizes an excessive number of partitions of a node into multiple children, thus accounting for the entropy of constructing multiple children nodes. Hence, even though Customer ID provides the maximum Information Gain = 1, it constructs 16 children (one for every unique value of Customer ID), each of which contains a single data instance. Having many small-size partitions can lead to overfitting since we would be learning the majority class at every leaf using only a few training instances, which can be highly biased by noise.

1. Consider the following 3 decision trees:





Tree 1 Tree 2



Tree3

Compute the difference between the entropy of overall data with the weighted entropy of the leaves for each of the three trees. Based on these differences, which tree would you choose for performing classification? Is the attribute chosen at the root of this tree the same as the attribute chosen for splitting in (e)? Briefly comment on the nature of your results, and the properties of the impurity measure used while constructing decision trees.

Difference in Entropy for Tree1 = 1

Difference in Entropy for Tree2 = 1

Difference in Entropy for Tree3 = 0.66

Looking at the difference between the entropy of the overall data and the weighted entropy at the leaves, we can observe that Tree1 and Tree2 provide the maximum difference (drop) in entropy. However, we have already seen in (e) that splitting on the Customer ID provides lower Gain Ratio than splitting on Housing Type, suggesting it to be a too complex model prone to overfitting. Hence Tree2 is preferred over Tree1 for performing classification. The attribute chosen at the root node is thus Marital Status, which is different than the attribute chosen in (e), which is Housing Type. This shows that even though using Marital Status at the root node does not show any improvement in performance initially, but by further splitting the children according to Gender, we can obtain an optimal tree that does a perfect job at classification (0 misclassifications). This shows that splitting on the basis of impurity measures is a greedy approach that may not always produce the optimal solution, especially in the scenarios where a true decision boundary involves interacting attributes. In this example, the two attributes Gender and Marital Status interact with each other to form a better optimal decision boundary.

Marital Status = Married, AND Gender = Female → C0

Marital Status = Single, AND Gender = Male → C0

Marital Status = Married, AND Gender = Male → C1

Marital Status = Single, AND Gender = Female → C1

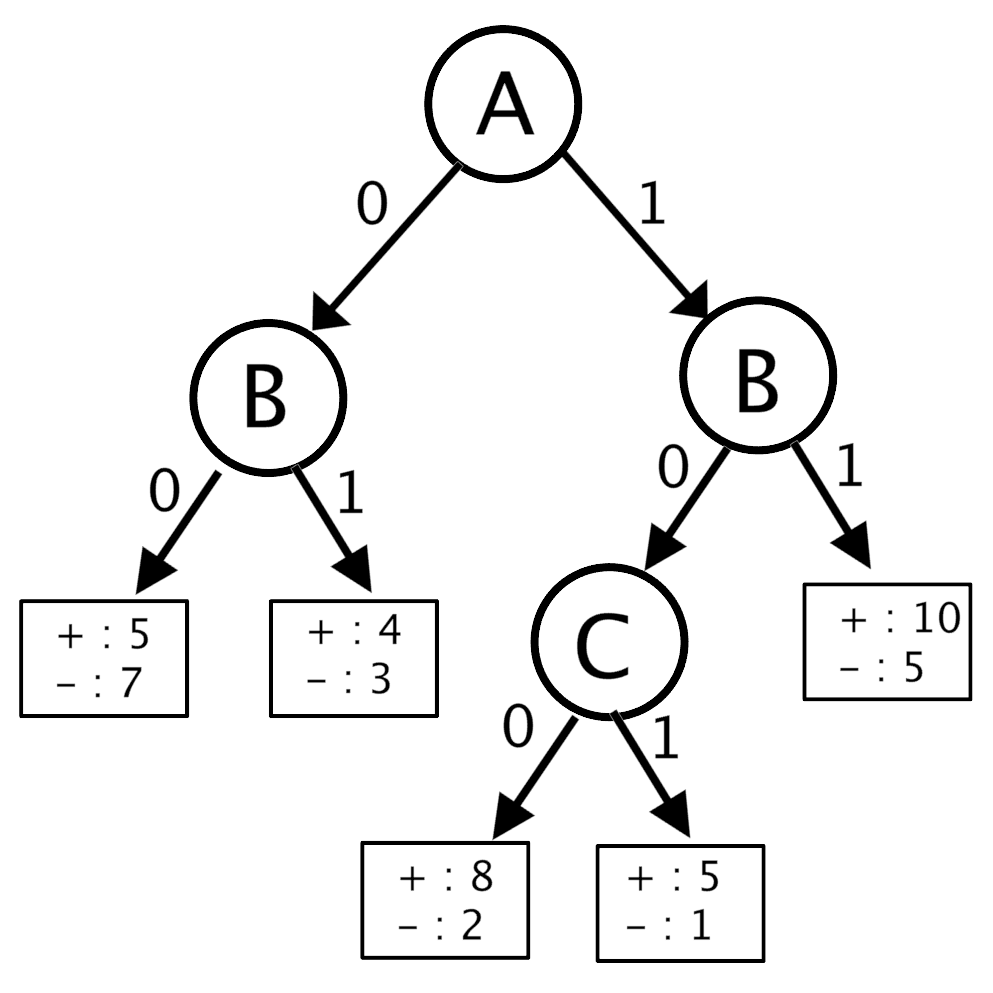
1. Consider a data set with four binary attributes X1, X2, X3 and X4. The attribute X4 takes the same value as X3 for each record, i.e., X4 is equal to X3. In the following scenario, find whether the decision boundary learnt by the two models would be similar; otherwise, find which of the two models would perform better. Provide a brief justification.

We build two decision trees:

a. T1, which is learnt using all the four attributes

b. T2, which is learnt using only three attributes X1, X2, and X3.

The performance of T1 and T2 would be exactly the same as the inclusion of attribute X4, which is identical to X3 would not alter the choice of splitting attribute at each node.

****

1. Using the decision tree above, how would you label (+ or -)  for the following?

1. A=0, B=0, C=1

Answer: +

1. A=1, B=1

Answer: +

1. A=0

Answer: -  (Because the value of B is unknown, we assign the majority class of all records with A=0)

1. What is the generalization error for the tree using the pessimistic error rate approach when the cost of each leaf is 3?

Generalization error = (opt error) + (penalty)\*(# leaves)/N

Optimistic error = (5+3+5+1+2)/50 = 16/50 = 0.32

→ Generalization error = (16/50) + (5\*3)/50 = 31/50 = 0.62

1. Consider the decision tree shown in the diagram below. The counts shown in the leaf nodes correspond to the number of training records associated with the nodes.

A close up of a watch

Description automatically generated

* 1. Using the above decision tree, what labels would you assign to the following records? (i) A = 0, B = 1, E = 0
     1. A = 1, C = 1
     2. A = 0, B = 1

(i) – (ii) + (iii) –

For (iii), the value of attribute E is unknown. In such a case, we should consider all the records with A = 0 and B = 1 and assign the label of the majority class. In this case, there are 8+2 = 10 records with + labels and 10+6 = 16 records with label -. Therefore, the label assigned is -.

* 1. Compute the training error rate for the tree.

The training error rate is 0.29.

* 1. Estimate the generalization error for the tree using the pessimistic error rate approach (Assume the cost of each leaf is 2).

The generalization error is 0.41

* 1. Suppose the nodes labeled as D and E in the previous decision tree are replaced by their corresponding leaf nodes so that leaf node corresponding to B = 0 has 20 (+) and 12(-) records and the leaf node corresponding to B = 1 has 10 (+) and 16(-) records. Estimate the generalization error of the pruned tree using the pessimistic error rate approach. Use your answers for part (b) and (c) to determine whether the original tree should be pruned.

The generalization error of the pruned tree is 0.4. Since pruning reduces the generalization error, the tree should be pruned.

* 1. Between the original decision tree and the pruned tree as indicated in part (d), which decision tree will you prefer if you are given the following validation set? Explain your choice.

A close up of a device

Description automatically generated

The error rate on the validation set before pruning is 0.3. The error rate on the validation set after pruning is 0.1. Since pruning reduces the error rate on the validation set, it is likely that the pruned tree would perform better on an unseen dataset and therefore, the pruned tree should be preferred.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** | **True class label** | **Original Tree** | **Pruned Tree** |
| 0 | 0 | 0 | 0 | 1 | + | + | + |
| 0 | 0 | 0 | 1 | 1 | + | - (mistake) | + |
| 0 | 0 | 1 | 1 | 1 | + | - (mistake) | + |
| 0 | 1 | 0 | 1 | 0 | - | - | - |
| 0 | 1 | 0 | 0 | 1 | + | + | -(mistake) |
| 0 | 1 | 1 | 1 | 1 | - | + (mistake) | - |
| 1 | 1 | 0 | 1 | 1 | - | - | - |
| 1 | 1 | 0 | 1 | 1 | - | - | - |
| 1 | 1 | 1 | 1 | 1 | + | + | + |
| 1 | 1 | 1 | 1 | 1 | + | + | + |

1. Consider the decision tree shown in the figure below, and the corresponding training and test sets in the tables below.

A close up of text on a white background

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Training set

A screenshot of a cell phone

Description automatically generated

Test set

1. Estimate the generalization error rate of the tree using both the optimistic approach and the pessimistic approach. While computing the error with a pessimistic approach, to account for model complexity, use a penalty value of 2 at each leaf node.

Error optimistic = 0; Error pessimistic = (0+6\*2)/15 = 0.8

1. Compute the error rate of the tree on the test set shown in the table above.

Test Error = (4+3+0+0+0+0) / 15 = 0.47

1. Compute the optimistic generalization error on the training set and the error on the test set when no pruning is done, and when pruning is done to get a pruned tree with 4 levels, 3 levels, and 2 levels. Note that the root node is at level 1. Comment on the behavior of training and test set errors with respect to model complexity on pruning from level 4 all the way up to level 2.

The answers are provided in the table below. As the tree is pruned more and more, the optimistic generalization error on the training set increases, but the error on the test set decreases to 0 when level 3 is pruned. This illustrates the overfitting problem at levels 3 and 4 of the tree. Pruning further from level 3 to level 2 leads to an increase in both training and test error, since the model becomes too simple and suffers from underfitting.

A screenshot of a cell phone

Description automatically generated

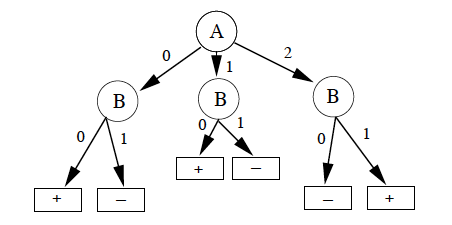
1. Using a penalty value of 2, compute the generalization error rate using the pessimistic approach on the pruned version of the original tree that has lowest error rate on the test set. Comment on the utility of incorporating model complexity in building a predictive model with respect to this question.

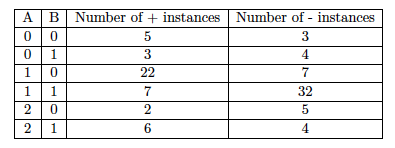
The lowest error rate on the test set is obtained for the pruned version of the tree with 3 levels.

Error pessimistic at level 3 = (2+4\*2)/ 15 = 2/3 = 0.666.

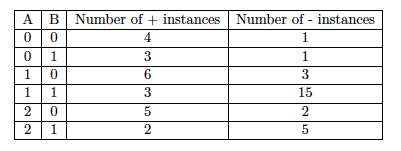
The original decision tree suffers from overfitting, which is reflected in the fact that its generalization error on the test set is significantly higher than that on the training set. Smaller trees are less likely to overfit the data. Indeed, the pruned tree’s pessimistic error rate is smaller than the pessimistic error rate of the original tree. The incorporation of model complexity through the pessimistic error rate allows a simpler (pruned) tree to be selected, even though it has a higher error rate on the training set.

1. Consider the decision tree shown in the figure below, and the corresponding training and test sets in the tables below.





Training set



Test set

1. Estimate the generalization error rate of the tree using both the optimistic approach and the pessimistic approach. While computing the error with the pessimistic approach, to account for model complexity, use a penalty value of 2 for each leaf node.

Optimistic Estimate = (3 + 3 + 7 + 7 + 2 + 4)/100 = 0.26

Pessimistic Estimate= [(3 + 3 + 7 + 7 + 2 + 4) + 2 x 6]/100 = 0.38

1. Compute the error rate of the tree on the test set shown in the table above.

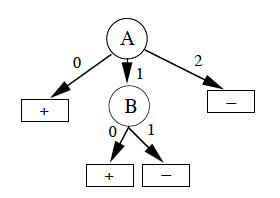
Test Error = (1 + 3 + 3 + 3 + 5 + 5)/50 = 0.4

1. The figure below shows a pruned version of the original decision tree. Estimate the generalization error rate of this tree using both the optimistic approach and the pessimistic approach, as in Part (a). Also, compute the error rate of this tree on the test set shown in the table above.

Optimistic Estimate = (7 + 7 + 7 + 8)/100 = 0.29

Pessimistic Estimate = [(7 + 7 + 7 + 8) + 2 x 4]/100 = 0.37

Test Error = (2 + 3 + 3 + 7)/50 = 0.3



Pruned version of tree

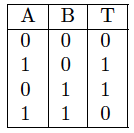
1. Use your answers of the pessimistic error rate approach from part (b) and (c) to determine whether the original tree should be pruned, and briefly explain.

Since pruning reduces the generalization error, the tree should be pruned.

1. Comment on the utility of incorporating model complexity in building a predictive model.

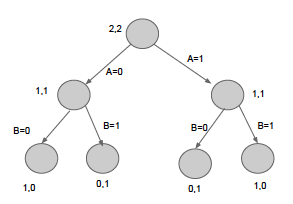
The original decision tree suffers from overfitting, which is reflected in the fact that its generalization error on the test set is significantly higher than that on the training set. Smaller trees are less likely to overfit the data. Indeed, the pruned tree’s error rate on both the training and test set are very similar. The incorporation of model complexity through the pessimistic error rate allows a simpler (pruned) tree to be selected, even though it has a higher error rate on the training set.

1. Consider the truth table for an XOR relationship between two dependent variables A and B and target variable T. Note that A, B and T are boolean variables.



(a) Draw a decision tree that expresses this truth table.

Answer: The x, y notation indicates the number of false records (x) and the number of true records (y) at each node. Note that for the leaf nodes, the figure was a bit truncated and so the comma looks like a period.



(b) Compute the information gain at each split in the decision tree drawn in (a).

Zero information gain at the first split and one at the second split.

(c) Consider a pruning condition: we stop recursion at a node once the information gain for all attributes at that node is zero. What is the decision tree formed if we use this pruning condition?

A single node with (2,2) samples of T = 0 or 1.

(d) Argue using the above example (or any other example of your choice) if the pruning condition mentioned in (c) is always a good heuristic for decision tree learning.

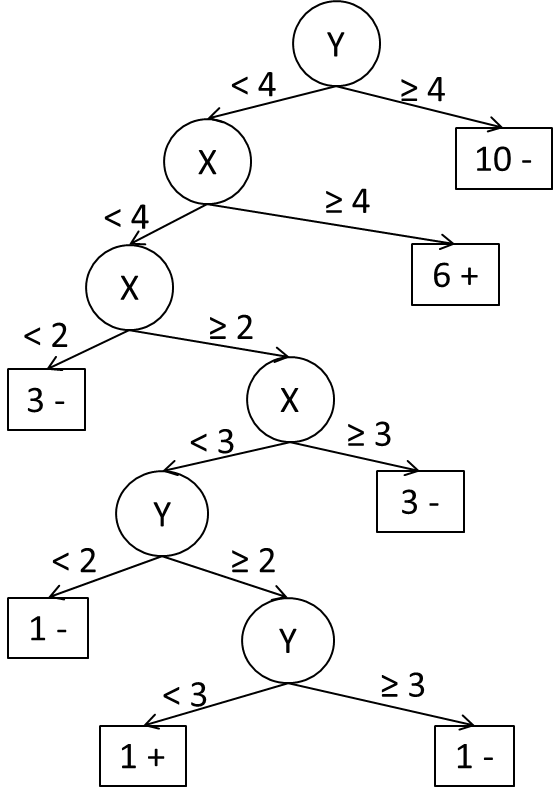
If we stop recursion once information gain for all attributes is zero, we may miss interesting patterns formed by combinations of attributes such as XOR.

1. Consider the training data given in the table below for classification, where the two classes of interest are ‘-’ and ‘+.’ We want to apply binary decision trees as our chosen algorithm for classifying this data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Y=5 | - | - | - | - | - |
| Y=4 | - | - | - | - | - |
| Y=3 | - | - | - | + | + |
| Y=2 | - | + | - | + | + |
| Y=1 | - | - | - | + | + |
|  |  |  |  |  |  |
|  | X=1 | X=2 | X=3 | X=4 | X=5 |
|  |  |  |  |  |  |

1. Find a decision tree that uses a minimum number of splits (decision boundaries at internal nodes) to perfectly classify each training data instance of the table above. Hint: The minimum number of splits that you need to create a perfect classifier is 6. You are ***not*** required to compute the Information Gain at each split for constructing the decision tree, but to arrive at your solution by visually inspecting the data.

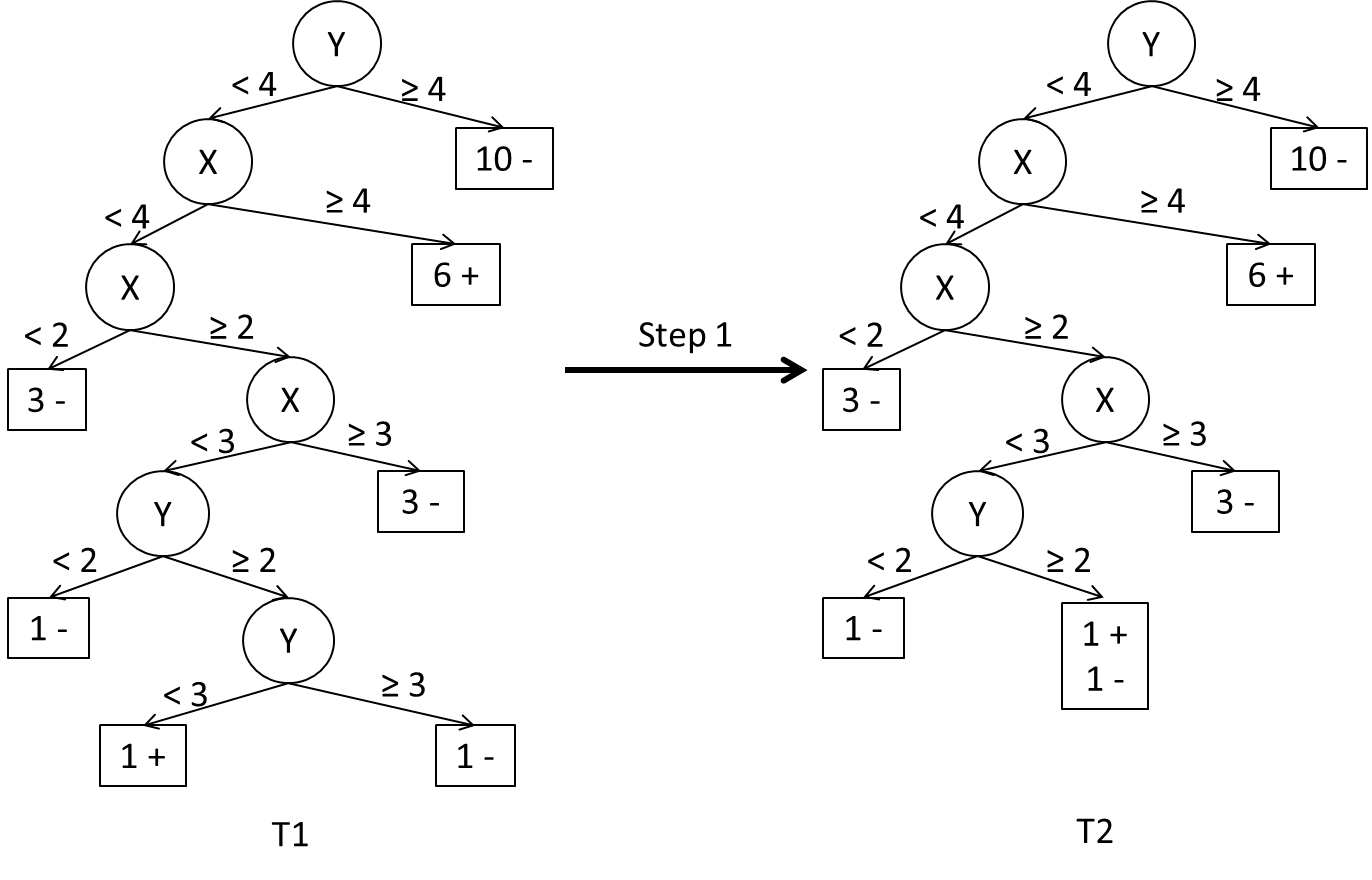
An example of a decision tree with 6 number of splits is given below.



There can be other variants of this tree with 6 internal nodes that can correctly classify each training instance. Any such tree would be accepted as a valid answer.

1. Use a pessimistic estimate of the generalization error to prune this tree using sub-tree replacement post-pruning method. Use Ω = 2 as the cost of adding a leaf node while calculating the pessimistic estimate. In case there is a tie in determining the majority class of a leaf node, use ‘-’ as the default majority class.

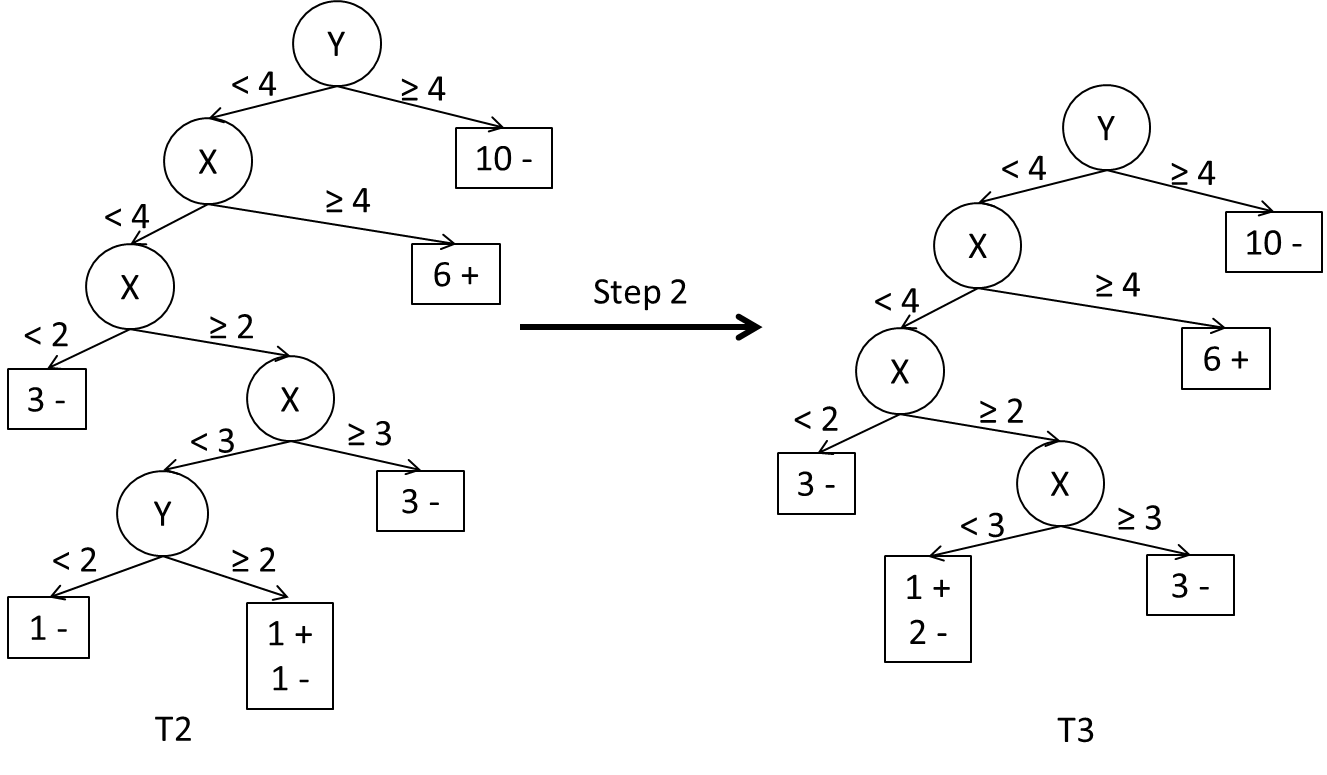
Using sub-tree replacement method, we choose a pair of leaf nodes in tree T and replace the subtree rooted at its parent with a leaf node, creating a pruned tree, Tpruned. If the pessimistic estimate of error reduces for Tpruned when compared with the original tree T, we replace T with Tpruned and recursively keep pruning T till the pessimistic estimate on Tpruned start exceeding the pessimistic estimate on T. We then output T as our pruned tree. The steps of this procedure on the tree shown in (c) can be shown as follows:



Pessimistic Estimate (T1) = (0 + 7x2)/25 = 14/25

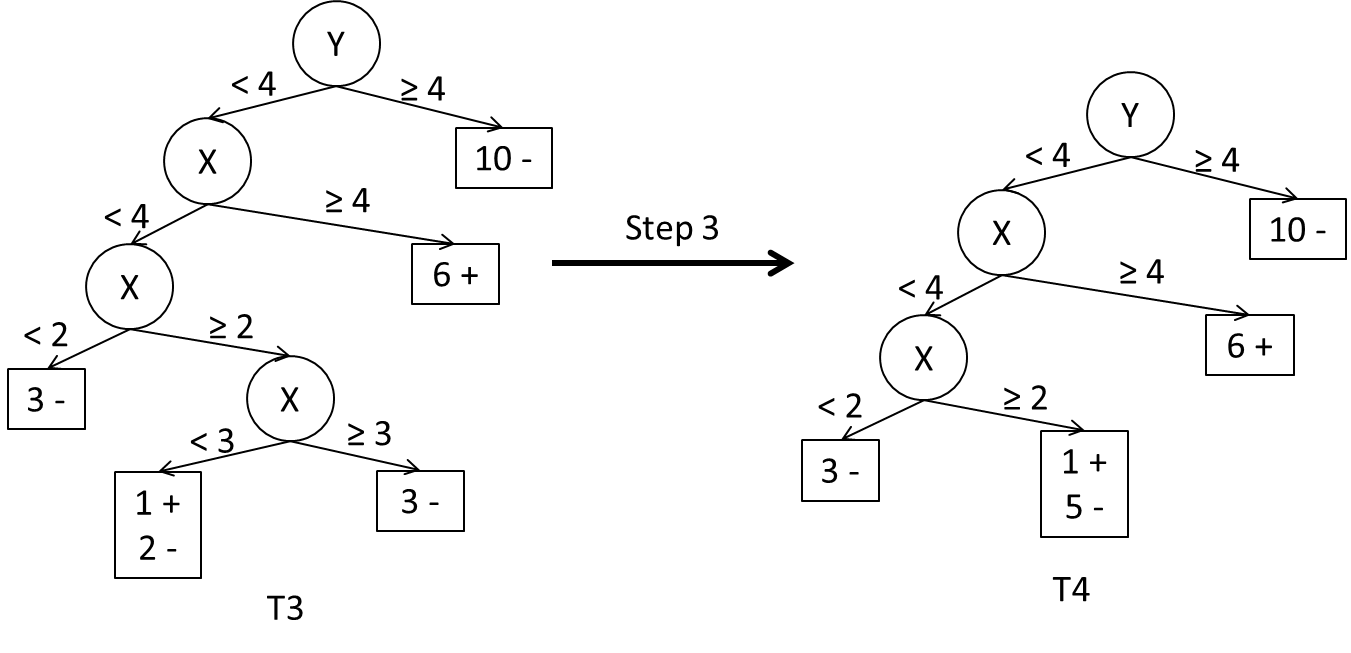
Pessimistic Estimate (T2) = (1 + 6x2)/25 = 13/25

Pessimistic Estimate(T2) < Pessimistic Estimate (T1), hence proceed further



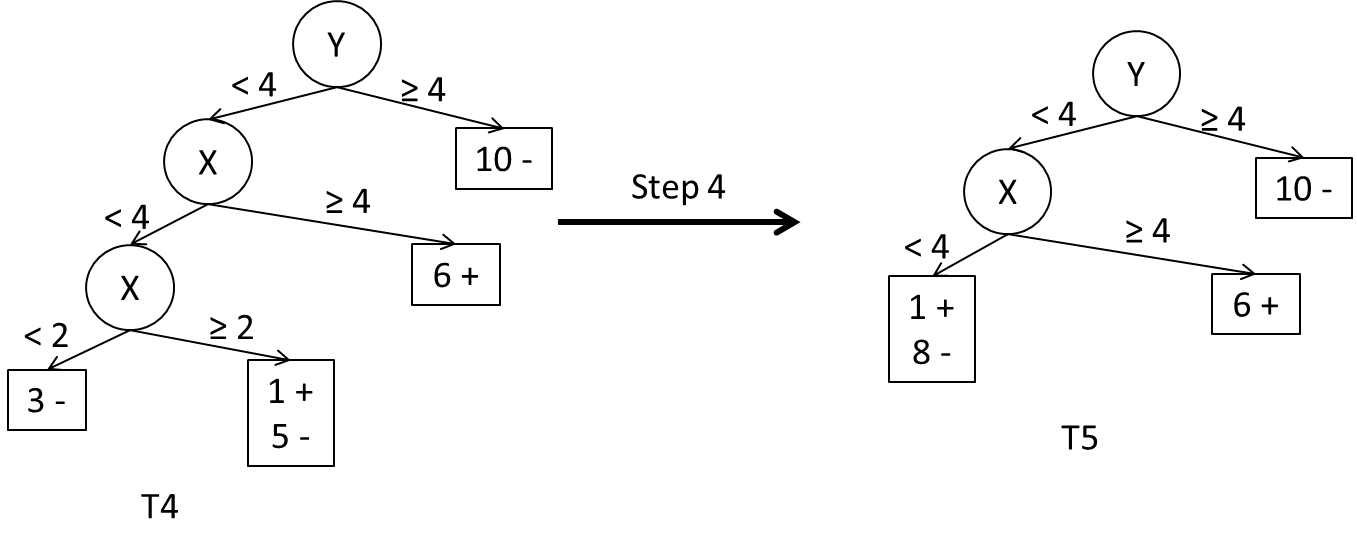
Pessimistic Estimate (T3) = (1 + 5x2)/25 = 11/25

Pessimistic Estimate(T3) < Pessimistic Estimate (T2), hence proceed further



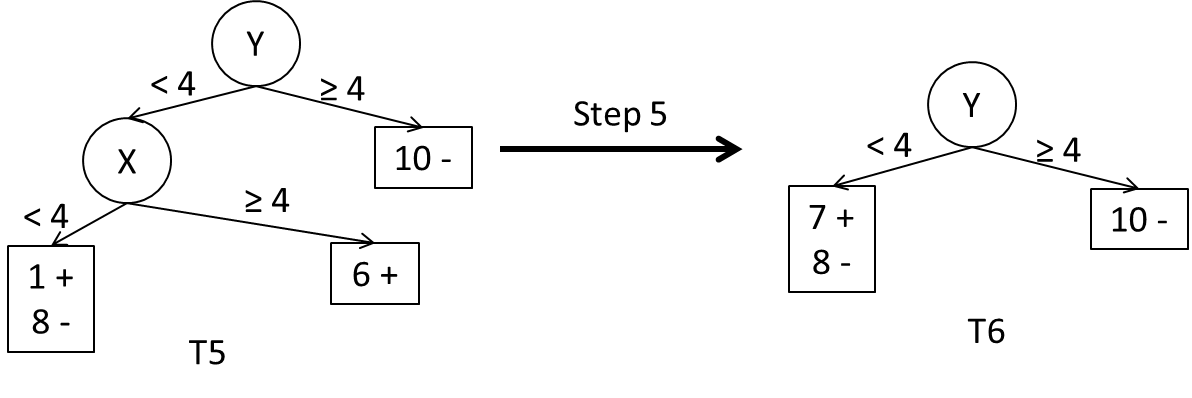
Pessimistic Estimate (T4) = (1 + 4x2)/25 = 9/25

Pessimistic Estimate(T4) < Pessimistic Estimate (T3), hence proceed further



Pessimistic Estimate (T5) = (1 + 3x2)/25 = 7/25

Pessimistic Estimate(T5) < Pessimistic Estimate (T4), hence proceed further



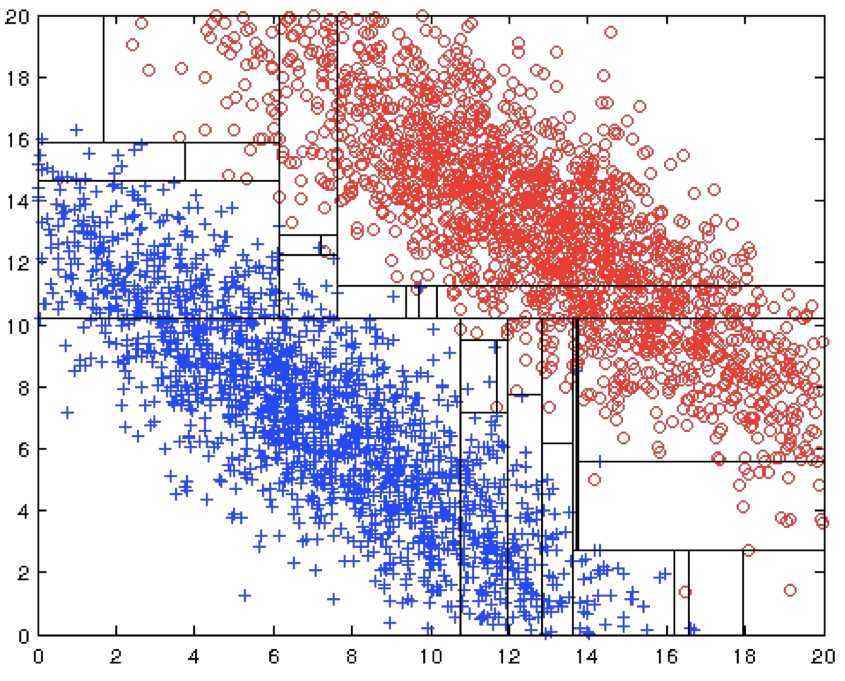
Pessimistic Estimate (T6) = (7 + 2x2)/25 = 11/25

Pessimistic Estimate(T6) > Pessimistic Estimate (T4), hence stop. T5 is our pruned tree.

1. Compare the nature of the original tree (with perfect classification on training data) with the pruned true by visual inspection. Would you use the original tree or the pruned tree for classifying any future instance of the data? State the name of the phenomena being explored in this question.

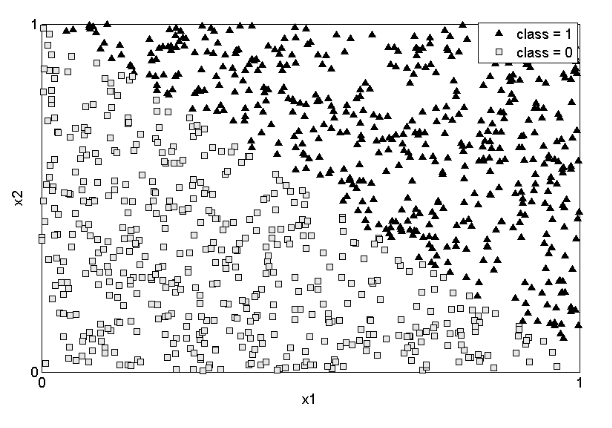
The original tree seems to be too complex given the pattern in the training data because of the presence of a noisy (+)-instance at (X = 2, Y = 2). By performing the subtree replacement pruning procedure, we obtain T5 as our pruned tree, which appears to be simpler than the original tree T1. This is captured in the fact that the optimistic error estimate of T1 is better than T5, but the pessimistic error estimate of T5 is better than T1. Hence, we would choose the pruned tree T5 for performing future classification. The phenomenon being explored in this question was “overfitting due to the presence of noise.”

1. Consider the two-dimensional data set in the figure below with two classes (circles and crosses). The lines on the figure correspond to the boundaries of a decision tree T1 learnt on this dataset. Suppose we add 50 additional irrelevant attributes to the above data and learn a decision tree T2 on the modified dataset. Comment on the expected performance of T2 relative to T1. Assume that the decision trees are built by choosing the attribute that provides maximum information gain on splitting at every internal node.

****

As shown in the figure, splitting on *X1* and *X2* can provide high enough gains at the shallower levels of the tree (in the first few splits). However, none of the attributes could individually provide substantial information gain at deeper levels of the tree and thus may appear similar to irrelevant attributes. As a result, the decision tree T2 is expected to include many irrelevant attributes for splitting at internal nodes once the tree becomes deep enough to achieve higher accuracy. Further, it is possible that X1 and X2 are not picked up at all even after performing a fairly large number of splits in T2, thus failing miserably on test data. This problem would become less severe if there were a larger number of training samples. This example illustrates the interplay between model complexity, number of irrelevant attributes, and interactions among the attributes. Note that if there were no irrelevant attributes, it would be possible to build a sufficiently complex decision tree T1 that shows good generalization performance using a smaller number of training examples than that required by T2.

1. Consider the following two-dimensional data set shown in the figure below with two classes (1 and 0).



Suppose that all these samples are used to train a decision tree, and the constructed tree has close to zero error on the training set. At least how many nodes are needed in the tree? Briefly explain.

a) 1

b) 2

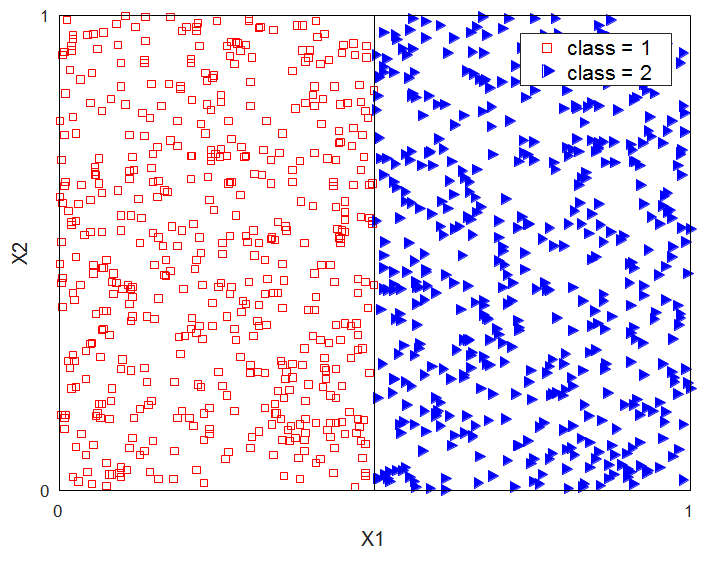
c) 4

d) many more than 5

d), since the decision tree has to use many splits to cover the non-horizontal and non-vertical class boundary.

1. Decision Trees and Irrelevant Attributes

Consider the following scenario shown in the figure below that has 1000 samples, half of which belongs to class 1 (shown as red squares), and the remaining half belong to class 2 (shown as filled blue triangles). Samples for these classes are distributed in the two-dimensional space defined by attributes X1 and X2 as shown in the figure. Answer the following questions with a brief explanation.



**Scenario 1**

1. Suppose all the 1000 samples in the dataset are used to train a decision tree that uses a single attribute to split at each node. If the constructed tree has close to zero error on the training set, then how many nodes will be needed to obtain such a tree? Briefly explain.

a) 1

b) 2

c) 4

d) many more than 10

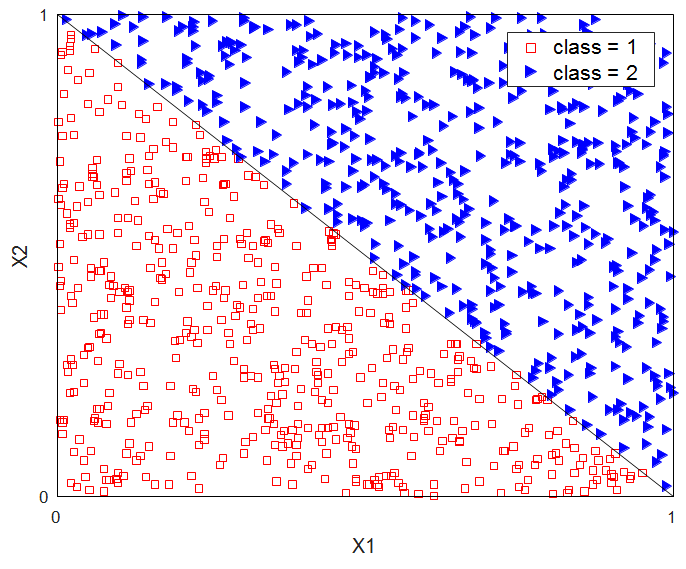
a) The decision tree would be able to separate samples from the two classes by making a split, X1<0.5 and X1≥0.5.

1. Now, suppose we add 100 additional attributes X3, X4, ..., X102 to the dataset such that samples from each of the two classes are uniformly distributed along each of these attributes (like X2). Comment on the performance of the decision tree learnt using such a dataset.

The decision tree constructed would not be impacted by the irrelevant attributes since they will all have poor information gain compared to X1. As a result, a decision tree constructed by splitting X1 at 0.5 would still be able to separate all the samples of the two classes.

Now consider another scenario shown in the figure below that has 1000 samples, half of which belong to class 1 (shown as red squares) and the remaining half belong to class 2 (shown as filled blue triangles). Samples for these classes are distributed in the two-dimensional space defined by attributes X1 and X2 as shown in the figure below. Answer the following questions with a brief explanation.

.



**Scenario 2**

**III.** Suppose all the 1000 samples in the dataset above are used to train a decision tree that uses a single attribute to split at each node. If the constructed tree has close to zero error on the training set, then how many nodes will be needed to obtain such a tree? Briefly explain.

1. 1
2. 2
3. 4
4. Many more than 5

d) since the decision tree must use many splits to cover the non-horizontal and non-vertical class boundary

**IV.** Now, suppose we add 100 additional attributes X3, X4, ..., X102 to the dataset such that samples from each of the two classes are uniformly distributed along each of these attributes. Comment on the performance of the decision tree learnt using such a dataset.

Although, *X1* and *X2* together can separate the two classes at the shallower levels of the tree, at deeper levels, none of the attributes could individually obtain substantial information gain on splitting and appear similar to noisy attributes. As a result, the decision tree is expected to include irrelevant attributes for splitting, would learn incorrect boundaries, and would fail on testing data.

We next present some of the most common mistakes, along with our explanation on what is wrong with each of these answers:

* Model complexity will increase significantly because of the additional 100 attributes. Not in a practical sense since the model complexity (in terms of the depth of the tree and the number of nodes) was already high in III, where we had only X1 and X2. However, the model complexity can be said to increase from an MDL perspective (if we consider the dimensionality of the data into account while computing model complexity).
* Too low training error would lead to overfitting. Not always! For example, in I, the training error is almost 0, yet there is no overfitting.
* Generalization error would be low as the tree is too complex. This is in general true, but not the only reason behind poor performance here. The impact of an additional 100 irrelevant attributes needs to be explained here. Note that the tree learnt in III is also complex, yet it would perform much better compared to the tree learnt in IV, because the latter would incorporate irrelevant attributes, whereas the tree learnt in III would only split using X1 or X2 and would perform better. The tree learnt in III could potentially have high generalization error, had there been noise in the training data, which doesn’t seem to be the case here, as there is a clear diagonal boundary that separates the two classes.

1. Choosing the right attributes for splitting in Decision Trees

Given a training dataset that contains 100 records of three attributes with their corresponding class labels, the goal is to learn a decision tree using the given dataset. For each attribute, the following information is given in the tables below: a) The number of possible values that an attribute can attain and b) the information gain achieved on splitting a root node using the given attribute. You need to choose one of the following three attributes to split at the root node of the decision tree.

(i)

|  |  |  |
| --- | --- | --- |
| **Attribute** | Number of possible values that an attribute can take | Information Gain  (IG) |
| A1 | 2 | 0.6 |
| A2 | 2 | 0.8 |
| A3 | 2 | 1 |

Attribute A3, since it has the highest information gain.

(ii)

|  |  |  |
| --- | --- | --- |
| **Attribute** | Number of possible values that an attribute can take | Information Gain  (IG) |
| A1 | 2 | 0.6 |
| A2 | 5 | 0.61 |
| A3 | 100 | 1 |

A1. Although Information Gain (IG) in A2 is slightly higher than A1, it splits into 5 nodes as opposed to 2 nodes for A1 and therefore, would learn a more complex tree compared to A1. A3 has a higher information gain than both A1 and A2, but looks to be an ID attribute and thus, a tree built on A3 would not generalize.

# **Model Evaluation and Selection**

1. Amazon announces a hiring challenge for building the best classification model that can predict an item’s profitability (high or low) based on its various characteristics. Amazon provides everyone with a labeled dataset, A1, that can be used to build a classification model (e.g., using any of the techniques discussed in the class). In addition, Amazon allows participants in this competition to check the accuracy of their models on another data set, A2, for which labels are hidden from the participants. Participants can submit their labels for A2 to Amazon, and it returns the accuracy of these labels on A2 back to them. Thus, this evaluation on A2 can be used by the participants to evaluate various models they are developing before they choose one of them to submit as their final solution to the challenge. The A2 data set is much smaller than A1.

You choose to participate in the challenge, so you build a classifier N1 using A1. For evaluation, you run it on both A1 and A2 and note the accuracy. You are also able to come up with 2 other classifiers N2 and N3. You repeat the above training and evaluation process with them. This gives you a table of accuracies for these 3 classifiers as shown in the following table [Accuracy% = 100% - Error Rate]

|  |  |  |
| --- | --- | --- |
| Classifier | Accuracy on A1 | Accuracy on A2 |
| N1 | 78% | 70% |
| N2 | 80% | 68% |
| N3 | 83% | 65% |

1. Which one of these classifiers - N1, N2, N3 would you submit as your entry into the challenge and why?   
   You’ll want to submit N1 as it has the lowest validation error (on A2) of all 3, even though its training error (on A1) is the largest. Please note that N1 is not underfit just because it has the lowest training accuracy of the three.
2. You’ve recently been looking at some Deep Learning courses online and decide to develop a number of highly complex deep neural networks and choose one of them that has the highest accuracy on A2. For this model, N4, the accuracy on A1 and A2 is given in the following table:

|  |  |  |
| --- | --- | --- |
| N4 | 90% | 70% |

Should you submit this classifier instead of the earlier three? Why or why not?

No, even though the training error is the lowest of the 4 and the validation error is also the lowest, the same as N1, by **Occam’s Razor**, given two models of similar generalization errors, one should prefer the simpler model over the more complex model.

1. Which dataset(s) should Amazon use to evaluate and rank the final submission by each participant?
   1. A1, the larger dataset
   2. A2, the smaller but hidden dataset
   3. Both A1 and A2
   4. Neither A1 nor A2

Please state the reasons behind your choice.   
(iv) Neither. Amazon would use another secret dataset, called the test set that all participants are unaware of. They wouldn’t use A1, because all submissions would be trained using that dataset and will have a high bias towards it. They wouldn’t use A2 since even though its hidden and no submission can be directly trained on it, participants may have used it to calculate the validation errors to pick their best model, so indirectly the final model they submit might have some bias towards A2. So, they’ll need a third unseen and unused dataset A3, which can give them a completely unbiased score for each submission.

1. There’s a competition on the popular data science website Kaggle for building the best model that can predict a movie’s genre (action, thriller, romcom, etc.) based on its plot summary. Kaggle provides everyone with a labeled dataset - D1, that can be used to build a classification model (e.g., using any of the techniques discussed in the book). In addition, Kaggle allows participants in this competition to check the error rate of their models on another data set D2, for which labels are hidden from the participants. Participants can submit their labels for D2 to Kaggle, and it returns the error rate of these labels on D2 back to them. Thus, this evaluation on D2 can be used by the participants to evaluate various models they are developing before they choose one of them to submit as their final solution to the competition. D2 data set is much smaller than D1.

You choose to participate in the competition, so you build a classifier C1 using D1. For evaluation, you run it on both D1 and D2 and note the error rates. You are also able to come up with 2 other classifiers C2 and C3. You repeat the above training and evaluation process with them. This gives you a table of error rates for these 3 classifiers as shown in the following table.

|  |  |  |
| --- | --- | --- |
| Classifier | Error Rate on D1 | Error Rate on D2 |
| C1 | 15% | 40% |
| C2 | 25% | 35% |
| C3 | 20% | 38% |

1. Which one of these classifiers - C1, C2, C3 would you submit as your entry into the competition and why?   
   You’ll want to submit C2 as it has the lowest validation error (on D2) of all 3, even though its training error (on D1) is the largest. Please note that C2 is not underfit just because it has the lowest training accuracy of the three.
2. Suppose your friend helps you out and shows you how to write a number of highly complex deep neural networks and choose the one that has the lowest error rate on D2. For this model, C4, the error rate on D1 and D2 is given in the following table:

|  |  |  |
| --- | --- | --- |
| C4 | 5% | 35% |

Should you submit this classifier instead of the earlier three? Why or why not?

A: No, even though the training error is the lowest of the 4 and the validation error is also the lowest, same as C2, by **Occam’s Razor**, given two models of similar generalization errors, one should prefer the simpler model over the more complex model.

1. Which dataset(s) should Kaggle use to evaluate and rank the final submission by each participant?
   1. D1, the larger dataset
   2. D2, the smaller but hidden dataset
   3. Both D1 and D2
   4. Neither D1 or D2

Please state the reasons behind your choice.   
(iv) Neither. Kaggle would use another secret dataset, called the test set, that all participants are unaware of. They wouldn’t use D1, because all submissions would be trained using that dataset and will have a high bias towards it. They wouldn’t use D2 since, even though its hidden and no submission can be directly trained on it, participants may have used it to calculate the validation errors to pick their best model, so indirectly, the final model they submit might have some bias towards D2. So, they’ll need a third unseen and unused dataset D3, which can give them a completely unbiased score for each submission. This is why we use 3 separate datasets - train, validation and test as a best practice.

1. You are given a classification dataset with 150 instances, which has been partitioned into three subsets: dataset A with 70 instances, dataset B with 50 instances and dataset C with 30 instances. Dataset A is used for training, dataset B for validation (to estimate generalization error) and dataset C is held-out for testing. You are supposed to compare two classification models: Model 1 and Model 2. The error rate of the two classification models on dataset A and datatset B are shown in Table 1.

|  |  |  |
| --- | --- | --- |
| Error Rate | Dataset A (Training) | Dataset B (Validation) |
| Model 1 | 0.02 | 0.28 |
| Model 2 | 0.18 | 0.20 |

Table 1

1. Which classification model would you expect to perform better on Dataset C? Support your answer with a brief explanation.

Model 2. It can be seen that the training error is lower for Model 1 as compared to Model 2. However, the validation error is higher for Model 1 as compared to Model 2. Since the validation error is representative of the generalization performance of a model on unseen instances, Model 2 is expected to have better performance. Also note that Model 2 shows similar error for both datasets A and B, highlighting the fact that it is not overfitted to the training set, and thus its error on the validation data is likely to be closer the true generalization error (i.e., one can expect similar error on data set C). In contrast, Model 1 performs much worse on the validation set, indicating that it is overfitted to the data used for its training, and thus is likely to perform much worse on data sets that are different than the one used for training.

1. For this part, assume that Model 1 & 2 are unpruned and pruned decision trees, respectively. You have recently been looking at some Deep Learning courses online and decide to develop highly complex deep neural networks. For this model (Model 3) the error rate on A and B is given in the following table:

|  |  |  |
| --- | --- | --- |
| Error Rate | Dataset A (Training) | Dataset B (Validation) |
| Model 3 | 0.01 | 0.19 |

Table 2

Should you choose Model 3 instead of the earlier two? Why or why not?

1. No, even though the training error is the lowest and the validation error is also the lowest, by **Occam’s Razor**, given two models of similar generalization errors, one should prefer the simpler model over the more complex model.
2. The validation error for model 3 is way higher than its training error. So, it is less likely to generalize well over unseen instances.
3. You are given a classification dataset with 100 instances, which has been partitioned into two subsets, dataset A with 50 instances and dataset B with 50 instances. Dataset A is used for training and dataset B is used for testing. You are supposed to compare two classification models: Model 1, which is an unpruned decision tree, and Model 2, which is a pruned version of the decision tree. The accuracy of the two classification models on datasets A and B are shown in the table below.

|  |  |  |
| --- | --- | --- |
| Classification Accuracy | Dataset A | Dataset B |
| Model 1 | 0.98 | 0.72 |
| Model 2 | 0.82 | 0.8 |

1. Based on the accuracies shown in the table above, which classification model would you expect to have better performance on unseen instances? Support your answer with a brief explanation.

The training accuracy (accuracy on dataset A) is higher for Model 1 as compared to Model 2. However, the test accuracy (accuracy on dataset B) is lower for Model 1 as compared to Model 2. This indicates the presence of over-fitting in Model 1, which has learned a complex un-pruned decision tree that is specific to instances in dataset A but does not generalize well to test instances in dataset B, that have not been seen during training. On the other hand, Model 2 shows similar accuracies for both datasets A and B, highlighting the fact that the pruned decision tree trained using dataset A generalizes well over unseen test instances in dataset B, even though the training accuracy of Model 2 is lower than that of Model 1. Since the test accuracy is representative of the generalization performance of a model on unseen instances, Model 2 is expected to have better performance than Model 1.

1. Now, you tested Model 1 and Model 2 on the entire dataset (A + B) and found that the classification accuracy of Model 1 on dataset (A + B) is 0.85, whereas the classification accuracy of Model 2 on the dataset (A + B) is 0.81. Based on this new information and your observations from the table above, which classification model would you finally choose for classification? Provide a brief explanation.

The higher accuracy of Model 1 on the entire dataset (A + B) can be attributed to the higher accuracy of Model 1 on the training dataset (dataset A), as compared to Model 2. However, it is evident from Table 2 that Model 1 suffers from over-fitting as it shows higher training accuracy but poorer test accuracy in comparison with Model 2.

Hence, comparing the accuracy of a model on the entire dataset is not representative of the generalization performance of the model on unseen test instances, and is thus not meaningful. Even though the accuracy of Model 1 is higher than Model 2 on the entire dataset (A + B), Model 2 should be chosen for classification as it provides better performance on unseen test instances in dataset B.

1. Both Minimum Description Length (MDL) and the pessimistic error estimate are techniques used for incorporating model complexity into the loss function. State one similarity and one difference between them in the context of decision trees.

One similarity between the two techniques is that they are both add the complexity of the model to its loss function in order to derive an evaluation metric for the model. In addition, the model complexity in both cases is a function of the size of the decision tree.  
  
However, they are different, since MDL specifies a penalty for internal nodes and leaf nodes, while the pessimistic error estimate specifies a penalty for leaf nodes only. Hence, in MDL the cost of the model is also a function of the number of attributes (the cost of each internal node is a function of the number of attributes), while for the pessimistic error estimate, the cost of the model is independent of the number of attributes.

1. You are given the following three approaches for estimating the generalization error of decision trees while training the model:
2. Optimistic error estimate
3. Pessimistic error estimate
4. Minimum description length
5. If you have a large number of attributes, which would be the ***best*** approach for estimating generalization error and why?

MDL, since it can consider the number of attributes to be encoded at each decision node of the tree.

1. If you have a limited number of training instances, which would be the ***worst*** approach for estimating generalization error and why?

Optimistic error estimate would not account for the lack of representative data and can make the learned decision tree suffer from overfitting.

1. Assume a decision tree has been built to perfectly classify the training set, and now needs to be pruned to avoid overfitting. Pruning can be performed by estimating the generalization error of different pruned versions of the tree and choosing the one that has the lowest generalization error. Consider the following three methods for estimating the generalization error rate:

1) Optimistic (resubstitution estimate)  
2) Pessimistic

3) Error on a validation set

Compare and contrast the effectiveness of these three methods in addressing the overfitting issue.

The optimistic error estimation approach assumes that training data is a good representation of the overall data and thereby estimates the generalization error as the error on the training set. Since this approach does not take model complexity into account, it will select the decision tree that gives the lowest error on training data. Unless the training set is very large, the selected decision tree is likely to be overfitted to the training data and have poor generalization performance.

The pessimistic error approach also estimates the generalization performance using the training data except that it takes model complexity into account. The advantage of this approach over the optimistic approach is that it avoids generating overly complex decision trees that may overfit the training data.

The last approach uses a validation set for estimating the generalization error. In this approach, the original labeled data is divided into two parts, training and validation sets. The decision tree is built on the training set while the generalization error is estimated using the validation set, which consists of all the records that are unseen by the trained model. This approach can be used to select the pruned tree that minimizes error on the validation data set. This approach provides a better way to estimate generalization error, but it requires more labeled data so that we have sufficient data for training and validation.

1. Consider the decision trees shown in the figure below. Assume they are generated from a data set that contains 16 binary attributes and 3 classes, C1, C2, and C3. Compute the total description length of each decision tree according to the minimum description length principle.

A close up of a clock

Description automatically generated

* The total description length of a tree is given by:

**Cost**(**tree, data**) = **Cost**(**tree**) + **Cost**(**data**|**tree**)

* Each internal node of the tree is encoded by the ID of the splitting attribute. If there are **m** attributes, the cost of encoding each attribute is log**2(m)** bits.
* Each leaf is encoded using the ID of the class it is associated with. If there are **k** classes, the cost of encoding a class is log**2(k)** bits.
* **Cost**(**tree**) is the cost of encoding all the nodes in the tree. To simplify the computation, you can assume that the total cost of the tree is obtained by adding up the costs of encoding each internal node and each leaf node.
* **Cost**(**data**|**tree**) is encoded using the classification errors the tree commits on the training set. Each error is encoded by log**2(n)** bits, where **n** is the total number of training instances.

Which decision tree is better, according to the MDL principle?

Because there are 16 attributes, the cost for each internal node in the decision tree is:

A description...

Furthermore, because there are 3 classes, the cost for each leaf node is:

A description...

The cost for each misclassification error is log**2**(**n**).

The overall cost for the decision tree (a) is 2 \* 4 + 3 \* 2 + 7 \* log**2(n)** = 14 + 7 log**2(n)**

The overall cost for the decision tree (b) is 4\*4+5\*2+4\*log**2(n)** = 26+4 log**2 n**. According to the

MDL principle, tree (a) is better than (b) if **n <** 16 and is worse than (b) if **n >** 16.

1. Consider two decision trees, T1 and T2, where T2 is obtained by pruning some of the subtrees in T1. T1 is the most exhaustive tree that can be created using classification error as the splitting criterion. List all of the following statements that are always true for any given training and test sets and give a brief explanation.
2. Training error of T1 ≥ Training error of T2.
3. Training error of T1 ≤ Training error of T2.
4. Test error of T1 ≤ Test error of T2.
5. Test error of T1 ≥ Test error of T2.

ii only. T2 will fit the training set no better than T1, since T1 is the tree with the minimum classifier error on the training set, and some information is lost in the pruning of T1 to create T2. However, there is no guarantee on the test error of T1 being smaller or greater than T2 (either case can happen).

1. Consider two decision trees, T1 and T2, where T2 is obtained by pruning some of the subtrees in T1. T1 is created using classification error as the splitting criterion. Which of the following statements are always true?
   * 1. Training error of T1 ≤ Training error of T2.
     2. Test error of T1 ≤Test error of T2.

a only. T2 will fit the training set no better than T1, since some information is lost in the pruning. However, there is no guarantee on the test error of T1 being smaller or greater than T2.

1. Consider three decision trees, T1, T2 and T3, where T1 is created using misclassification error as the splitting criterion, T2 is obtained by pruning some of the subtrees in T1, and T3 is obtained by pruning some of the subtrees in T2. Note that pruning is performed to reduce the error on the validation set. Which of the following statements could be true? Briefly explain.
2. Training error of T1 > Training error of T2.

Not true. Pruning does not decrease the error rate on the training set. Pruned tree T2 will most likely have larger or unchanged training error than the original tree T1.

1. Validation error of T3 < Validation error of T2 < Validation error of T1

True. Since pruning is performed to reduce the validation error, validation error of T3 is better than that of T2, which is better than that of T1.

c) Test error of T3 ≤ Test error of T2 ≤ Test error of T1.

It depends. Generally, if the validation set is representative of the test set, then pruning T1 to T2 to T3 based on validation errors will lead to a reduction in test error as well (it will overcome the phenomenon of overfitting of T1). However, it is possible that validation error does not fully represent test error and pruning can lead to an underfitted tree that matches the validation set very well but misses some of the generalizable patterns in the data that appear in the training and test sets, leading to higher test errors.

1. The figure below shows three different decision tree classifiers built on a training sample of the same data set. The dataset consists of instances of class ‘X’ and instances of class ‘O.’ The decision boundary of the classifiers is indicated by the boundary line inside the rectangle and the classification decision is made as follows: everything above the boundary is classified as ‘O’ and everything below the boundary is classified as ‘X.’ Assume the future instances of the data are similar in terms of distribution and class composition.



1. (b) (c)

a) Which decision tree is the best fit for the training sample and why?

Model 1c is best fitted to the training data, as the number of misclassified instances, in this case, is the smallest.

b) Which model will best predict future data? Explain.

Model 1b is the best fit to future data. Since data is drawn from similar distribution (‘O’s above and ‘X’s below the diagonal), this model will most likely make best prediction on new data.

c) What are the phenomena in Figure 1a) and 1c) called?

Under-training and overfitting, respectively.

1. Consider a two-class problem, where there are equal numbers of positive and negative instances in the data. Suppose the class labels for the instances have been generated randomly (i.e., the attributes do not have any information about the class labels, and the accuracy of any classifier cannot be greater than 0.5). The classifier used is an un-pruned decision tree (i.e., a perfect memorizer with zero training errors). Consider the following cross-validation strategies for evaluation:
   1. The holdout method, where two-thirds of the data are used for training and the remaining one-third are used for testing.
   2. Ten-fold cross-validation.
   3. The .632 bootstrap method.

What will be the expected accuracy that will be estimated by each of the three cross-validation strategies for this problem? Based on these results, which cross-validation strategy provides the most inaccurate evaluation of the classifier’s performance for this problem?

Hint: While the .632 bootstrap approach is useful for obtaining a reliable estimate of model accuracy, it has a known limitation mentioned in [1], which is being highlighted in this problem.

[1] R. Kohavi. A Study on Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *In Proc. of the 15th Intl. Joint Conf. on Artificial Intelligence*, pages 1137–1145, Montreal, Canada, August 1995.

It is given that the test accuracy of any classifier on this dataset cannot be greater than

* 1. Further, since the classifier is an unpruned decision tree with zero training errors, its training accuracy is 1.

The accuracy estimate of holdout will thus be roughly equal to 0.5 on the one-third data used for testing. The accuracy of 10-fold cross validation on every test fold will be roughly equal to 0.5, providing an average test accuracy of 0.5 overall folds.

However, the accuracy estimate of .632 bootstrap over every bootstrap sample would be equal to (0.632 x test accuracy + 0.368 x training accuracy) = (0.632 x 0.5 + 0.368 x 1) = 0.684.

Since the actual test accuracy of the classifier is 0.5, both holdout and 10-fold cross-validation can provide reasonable estimates of the classifier’s performance. However, the accuracy estimate of .632 bootstrap is considerably different than the actual test accuracy of the classifier, and hence .632 bootstrap provides the most inaccurate evaluation of the classifier’s performance. This highlights a limitation of the .632 bootstrap specific to this problem.

# **Rule-Based Classifier**

1. Consider the problem of predicting if a given person is a defaulted borrower (DB) based on the following attribute values:

• Home Owner = Yes, No  
• Marital Status = Single, Married, Divorced   
• Annual Income = Low, Medium, High  
• Currently Employed = Yes, No

Suppose a rule-based classifier produces the following rules:   
• Home Owner = Yes --> DB = Yes

• Marital Status = Single --> DB = Yes

• Annual Income = Low --> DB = Yes

• Annual Income = High, Currently Employed = No --> DB = Yes

• Annual Income = Medium, Currently Employed = Yes --> DB = No

• Home Owner = No, Marital Status = Married --> DB = No

• Home Owner = No, Marital Status = Single --> DB = Yes

Answer the following questions. Make sure to provide a brief explanation or an example to illustrate the answer.

(a) Are the rules mutually exclusive?  
(b) Is the rule set exhaustive?  
(c) Is ordering needed for this set of rules?  
(d) Do you need a default class for the rule set?

Solution:

(a) No. The instance {Home Owner = Yes, Marital Status = Single} will trigger the first two rules.

(b) No. The instance {Marital Status = Divorced, Home Owner = No, Annual Income = High, Currently Employed = Yes} is not covered by any of the rules.

(c) Yes, because a record can match two or more rules that give conflicting predictions about the class. For example, the instance {HomeOwner=Yes, MaritalStatus=Divorced, AnnualIncome=Medium, CurrentlyEmployed=Yes} will trigger rule 1 (prediction: DefaultBorrower = Yes) and rule 5 (prediction: DefaultBorrower = No). If you do not tell the system to prefer one rule to another (i.e., order them), the system will not know how to classify the instance.

(d) Yes, since the rules are not exhaustive.

1. Consider the problem of predicting whether a given person will be a profitable customer (PC) for a retail store based on the following attributes:

* Annual income = low, medium, high
* Number of credit cards = less than 4, 5-10, 11 or more
* Purchases accessories = Yes, No
* Purchases extended warranties = Yes, No

Suppose a rule-based classifier produces the following rules:

* Purchases extended warranties = Yes → PC = Yes
* Annual income = medium → PC = Yes
* Number of credit cards = less than 4 → PC = No
* Purchases accessories = Yes, Annual income = high → PC = Yes
* Purchases extended warranties = No, Annual income = medium → PC = No

Answer the following questions with an explanation or example to support your answer.

1. Are the rules mutually exclusive?

No. The instance {Purchases extended warranties=Yes, Annual income = medium} will trigger the first two rules.

1. Is ordering needed for this set of rules?

Yes, because a record can match two or more rules that give conflicting predictions about the class. For example, the instance {Annual income=medium, Number of credit cards=less than 4} will trigger Annual income = medium → PC = Yes and Number of credit cards = less than 4 → PC = No. If you do not tell the system to give preference to one rule over another, the system will not know how to classify them.

1. Do you need a default class for the rule set? Why or why not?

Yes, because the rules are not exhaustive.

1. Consider the following set of rules based on the vertebrate data set given in Table 3.2 in the textbook.

R1: (Gives Birth = No, Aerial Creature = Yes) => Birds  
 R2: (Gives Birth = No, Aquatic Creature = Yes) => Fishes  
 R3: (Body Temperature = warm-blooded)  => Mammals  
 R4: (Gives Birth = No, Aerial Creature = No, Aquatic Creature = No) => Reptiles  
  
For each of the following questions, provide a yes/no answer with a short explanation or a suitable example to justify your answer.

1. Are the rules mutually exclusive?

No.   
Example: Pigeons satisfy both R1 and R3.

1. Are the rules exhaustive?

No.   
Example: The instance ‘salamander’ doesn’t satisfy any of the given rules.

1. Is ordering needed for this set of rules?

Yes.   
Example: R1 needs to come before R3. Say we interchange R1 and R3. Pigeons will then get classified as mammals.

1. For the data set below, build a rule set that classifies all instances perfectly. The rules are RIPPER-style rules, operators permitted are ≤, ≥, = and AND. An example rule is Y≥4 AND X=3 ⇒ -.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Y=5 | - | - | - | - | - |
| Y=4 | - | - | - | - | - |
| Y=3 | + | + | + | + | + |
| Y=2 | - | - | - | + | + |
| Y=1 | - | - | - | + | + |
|  | X=1 | X=2 | X=3 | X=4 | X=5 |

We provide a partial rule set  
Y≥4 ⇒ -  
Y=3 ⇒ +.

Add two more rules to this rule set with as few conditions as possible such that

1. Perfect classification can be attained without rule or class ordering,

Y≥4 ⇒ -

Y=3 ⇒ +

Y≤2 AND X≤3 ⇒ -

Y≤2 AND X≥4 ⇒ +

1. Perfect classification can be attained with rule ordering but without class ordering.

Y≥4 ⇒ -

Y=3 ⇒ +

X≤3 ⇒ -

X≥4 ⇒ +

1. Assume a data set of 2 attributes X and Y, each taking discrete ordinal values between 1 and 5.

There are 2 classes ‘+’ and ‘-’. Some hypothetical rule generation algorithm generated the following rule set.

|  |  |  |
| --- | --- | --- |
| *R*1 | : | *Y* ≤ 4 ∧ *Y >* 2 ∧ *X* ≤ 2 ⇒ + |
| *R*2 | : | *Y* ≤ 4 ∧ *Y >* 2 ∧ *X >* 3 ⇒ + |
| *R*3 | : | − |

We are also given the following table that shows the class label for the validation rule set and the predicted labels based on the original rules set or the rule set with modifications to R1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Condition | *Class Label for validation*  *set* | *Unmodified*  *R1* | *Y <*= 4 removed | *Y >* 2 removed | *X <*= 2 removed |
| Y = 4 ^ X = 1 | + | + | + | + | + |
| Y = 4 ^ X = 2 | + | + | + | + | + |
| Y = 4 ^ X = 3 | + | − | − | − | + |
| Y = 4 ^ X = 4 | + | + | + | + | + |
| Y = 1 ^ X = 1 | − | − | − | + | − |
| Y = 5 ^ X = 1 | − | − | + | − | − |

1. What is the accuracy, precision and recall (with respect to the ‘+’ class) of the rule set on the validation set?

Accuracy = 5/6

Precision = 3/3

Recall = 3/4

1. Optimize R1 using the following method: for every condition Ci in R1, create a new rule set by removing that condition (that is, let the new rule set be {R1 − Ci, R2, R3}).
2. Show the confusion matrix for this rule set.

Confusion Matrix for the following categories is as shown below:

|  |  |  |
| --- | --- | --- |
|  | + | - |
| + | 3 | 1 |
| - | 0 | 2 |

* 1. Unmodified:

|  |  |  |
| --- | --- | --- |
|  | + | - |
| + | 3 | 1 |
| - | 1 | 1 |

* 1. Y <= 4 removed:

|  |  |  |
| --- | --- | --- |
|  | + | - |
| + | 3 | 1 |
| - | 1 | 1 |

* 1. Y > 2 removed:

|  |  |  |
| --- | --- | --- |
|  | + | - |
| + | 4 | 0 |
| - | 0 | 2 |

* 1. X <= 2 removed:

1. Calculate the accuracy, precision and recall (with respect to the ’+’ class).
2. Unmodified:

Accuracy = 5/6 Precision = 3/3 Recall = 3/4

1. Y <= 4 removed:

Accuracy = 4/6 Precision = 3/4 Recall = 3/4

1. Y > 2 removed:

Accuracy = 4/6 Precision = 3/4 Recall = 3/4

1. X <= 2 removed:

Accuracy = 1 Precision = 1 Recall = 1

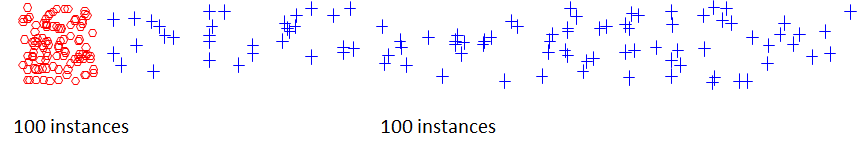
1. Decide whether our algorithm would keep this ’optimized’ rule or whether it would keep the original rule.

Answer: Yes, our algorithm would keep R1: Y <= 4 ^ Y > 2 ==> + optimized rule as the accuracy improves for the pruned rule and it also did not result in additional false predictions.

1. Give an explanation of what may have caused the hypothetical rule set generator to discover the suboptimal rule R1.

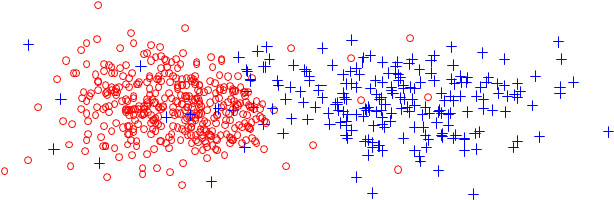
Answer: There could have been a condition such as Y = 4|3 ^ X = 3 for class - which acted as noise and this may have caused the hypothetical rule set generator to discover the suboptimal rule R1.

# **Nearest Neighbor Classifier**

* + - 1. For each of the two given scenarios, make a right choice of K for the KNN classifier in order to obtain better performance with a brief explanation.

**(a)** K = 1 or K=5 or K = 50?

K =1 would work the best. K = 5 might lead to misclassification of boundary points. K = 50 would lead to misclassification of a large number of blue points that are nearer to the boundary due to the high density of the red class.



**(b)** K = 1 or K = 5 or K=50?

K =1 would lead to misclassification of points due to the presence of some of the noisy points. For example, all the red points that are close to the noisy blue points on the left side could get misclassified as blue. This issue could be resolved by setting K to higher values, such as K = 5. However, setting. K to extremely high values (such as K = 50) could lead to misclassification of a large number of points belonging to the class that has a lower density (in this case, the blue class). For example in the above case, at K = 50, most of the points belonging to the blue class that are situated closer towards the heap of red class may get wrongly classified to red because they are likely to have more red neighbors due to the high density of the red class.

# **Naïve Bayes Classifier**

1. Consider the data set shown in the Table below.

A screen shot of a computer

Description automatically generated

1. Estimate the conditional probabilities for **P**(**A** = 1|+), **P**(**B** = 1|+), **P**(**C** = 1|+), **P**(**A** = 1|−),   
   **P**(**B** = 1|−), and **P**(**C** = 1|−).

P(A = 1|+) 0.6

P(B = 1|+) 0.4

P(C = 1|+) 0.8

P(A = 1|−) 0.4

P(B = 1|−) 0.4

P(C = 1|−). 0.2

1. Use the conditional probabilities in part (a) to predict the class label for a test sample (**A** = 1**, B** = 1**, C** = 1) using the naïve Bayes approach.

Let R : (A = 1,B = 1,C = 1) be the test record. To determine its class, we need to compute P(+|R) and P(−|R). Using Bayes theorem, P(+|R) = P(R|+)P(+)/P(R) and P(−|R) = P(R|−)P(−)/P(R). Since P(+) = P(−) = 0.5 and P(R) is constant, R can be classified by comparing P(R|+) and P(R|−).

1. For this question,
2. P(R|+) = P(A = 1|+) . P(B = 1|+) . P(C = 1|+) = 0.192
3. P(R|−) = P(A = 1|−) . P(B = 1|−) . P(C = 1|−) = 0.032
4. Since P(R|+) is larger, the record is assigned to (+) class.
5. Compare **P**(**A** = 1), **P**(**B** = 1), and **P**(**A** = 1**,B** = 1). State the relationships between **A** and **B**.

P(A = 1) = 0.5, P(B = 1) = 0.4 and P(A = 1,B = 1) = 0.2 = P(A=1)×P(B=1). Therefore, A and B are

independent.

1. Repeat the analysis in part (c) using **P**(**A** = 1), **P**(**B** = 0), and **P**(**A** = 1**,B** = 0).

P(A = 1) = 0.5, P(B = 0) = 0.6, and P(A = 1,B = 0) = 0.3 = P(A = 1) × P(B = 0). A and B are still

independent.

1. Compare **P**(**A** = 1**,B** = 1|**Class** = +) against **P**(**A** = 1|**Class** = +) and **P**(**B** = 1|**Class** = +). Are the variables conditionally independent given the class?

Compare P(A = 1,B = 1|+) = 0.2 against P(A = 1|+) = 0.6 and P(B = 1| +) = 0.4. Since the product of P(A = 1|+) and P(A = 1|−) are not the same as P(A = 1,B = 1|+), A and B are not conditionally independent given the class.

1. Consider the dataset shown in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instance | A | B | C | Class |
|  |  |  |  |  |
| 1 | 0 | 0 | 1 | - |
|  |  |  |  |  |
| 2 | 1 | 0 | 1 | + |
|  |  |  |  |  |
| 3 | 0 | 1 | 0 | - |
|  |  |  |  |  |
| 4 | 1 | 0 | 0 | - |
|  |  |  |  |  |
| 5 | 1 | 0 | 1 | + |
|  |  |  |  |  |
| 6 | 0 | 0 | 1 | + |
|  |  |  |  |  |
| 7 | 1 | 1 | 0 | - |
|  |  |  |  |  |
| 8 | 0 | 0 | 0 | - |
|  |  |  |  |  |
| 9 | 0 | 1 | 0 | + |
|  |  |  |  |  |
| 10 | 1 | 1 | 1 | + |
|  |  |  |  |  |
|  |  |  |  |  |

1. Estimate the conditional probabilities for P(A = 1|+), P(B = 1|+), P(C = 1|+), P(A = 1|−),

P(B = 1|−), and P(C = 1|−).

P(A = 1|+) = 0.6, P(B = 1|+) = 0.4, P(C = 1|+) = 0.8, P(A = 1|−) = 0.4, P(B = 1|−) = 0.4 and P(C = 1|−) = 0.2

1. Use the conditional probabilities in part (a) to predict the class label for a test sample (A = 1, B = 1, C = 1) using the naïve Bayes approach.

Let R : (A = 1,B = 1,C = 1) be the test record. To determine its class, we need to compute P(+|R) and P(−|R). Using Bayes theorem, P(+|R) = P(R|+)P(+)/P(R) and P(−|R) = P(R|−)P(−)/P(R). Since P(+) = P(−) = 0.5 and P(R) is constant, R can be classified by comparing P(R|+) and P(R|−).

For this question,

P(R|+) = P(A = 1|+) \* P(B = 1|+) \* P(C = 1|+) = 0.192

P(R|−) = P(A = 1|−) \* P(B = 1|−) \* P(C = 1|−) = 0.032

Since P(R|+) is larger, the record is assigned to (+) class.

1. Compare P(A = 1, B = 1|Class = +) against P(A = 1|Class = +) and P(B = 1|Class = +). Are the variables conditionally independent given the class?

P(A = 1|+) = 0.6, P(B = 1|+) = 0.4 and P(A = 1,B = 1|+) = 0.2. Therefore, A and B are not conditionally independent.

1. Let us consider the data instance (A=1, B=1, C=1). Using the naïve Bayes approach, compute the probability of this instance belonging to Class = + using
   1. no attributes (i.e., calculate prior probability)
   2. attribute A [ P(Class = +|A=1) ]
   3. attributes A and B [ P(Class = +|A=1, B=1) ]
   4. attributes A, B and C [ P(Class = +|A=1, B=1, C=1) ]

Comment on the change in probability values as we proceed from (i) to (iv).

By using naïve Bayes approach, we assume the variables are conditional independent.

i) P(+) = 0.5

ii) P(+ | A = 1) = = = 0.6

iii) P(+ | A = 1, B=1) = = = 0.6

1. P(+ | A = 1, B=1, C=1) = = =0.857.

Note that in the above calculations, we used Naïve Bayes to compute the denominator term of the Bayes equation. For example, to obtain P(A=1, B=1), we evaluated P(A=1, B = 1|+)P(+) + P(A=1, B = 1|+)P(+), where the conditional probabilities were computed using Naïve Bayes assumption. if we use Bayes theorem to calculate P(A=1,B=1,C=1). This ensures that the sum of the posterior probabilities over both classes equals 1. One may consider an alternate approach to compute the denominator term P(A = 1, B = 1) by directly counting its occurrences in the table. However, this approach may not always result in posterior probabilities that sum to 1 and thus must be avoided.

Comment: As more and more information/attributes is/are available, the classifier is getting more certain about labeling the given instance.

1. Consider the dataset shown in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instance | A | B | C | Class |
|  |  |  |  |  |
| 1 | 0 | 0 | 1 | - |
|  |  |  |  |  |
| 2 | 0 | 0 | 1 | + |
|  |  |  |  |  |
| 3 | 0 | 0 | 0 | - |
|  |  |  |  |  |
| 4 | 1 | 0 | 0 | - |
|  |  |  |  |  |
| 5 | 0 | 0 | 1 | + |
|  |  |  |  |  |
| 6 | 0 | 0 | 1 | + |
|  |  |  |  |  |
| 7 | 1 | 0 | 0 | - |
|  |  |  |  |  |
| 8 | 0 | 0 | 0 | - |
|  |  |  |  |  |
| 9 | 0 | 1 | 0 | + |
|  |  |  |  |  |
| 10 | 0 | 1 | 1 | + |
|  |  |  |  |  |

1. Estimate the conditional probabilities for P(A = 1|+), P(B = 1|+), P(C = 1|+), P(A = 1|−), P(B = 1|−), and P(C = 1|−) using the table given above.

P(A = 1|+) = 0, P(B = 1|+) = 0.4, P(C = 1|+) = 0.8, P(A = 1|−) = 0.4, P(B = 1|−) = 0 and P(C = 1|−) = 0.2

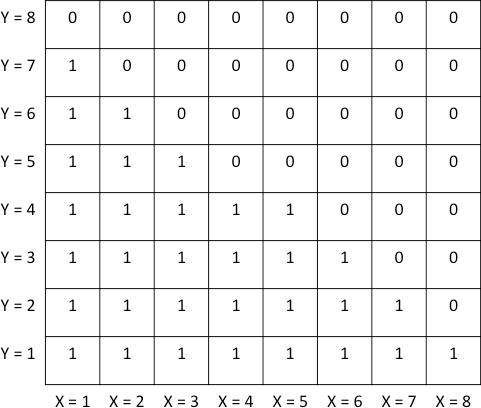
1. For a new data instance, **x** = (A = 1, B = 1, C = 1), compute the posterior probabilities, P(+|x) and P(-|x) using the naïve Bayes approach on the table given above.

P(+|x) = 0/0 and P(-|x) = 0/0. Thus, we cannot assign a class to this instance

1. What kind of problems will you encounter in predicting the class of **x** using the posterior probabilities computed in (f), and how can you resolve them?

If a conditional probability is 0, then the complete posterior probability becomes zero. If this happens for both classes, then it is not possible to assign a class to the instance. To fix this problem, we can use alternate ways of calculating posteriors such as m-estimate and Laplace.

1. Consider a two-dimensional dataset for classification, represented as a table in the figure below, where X and Y are the two attributes, each taking 8 discrete values. Each cell in the table corresponds to a data instance, and the value in the cell (0 or 1) represents its class label. For example, the data instance: (X = 2, Y = 1) belongs to class 1. There are an equal number of 0s and 1s in the dataset.



Evaluate the following statements about the performance of the naïve Bayes algorithm on this dataset, and indicate whether they are true or false:

1. Naïve Bayes would have no difficulty in classifying the data instance: (X = 8, Y = 2).

P(0 | X = 8, Y = 2) = P(X = 8 | 0) x P(Y = 2 | 0) x P(0) / P(X = 8, Y = 2) = (7/32) x (1/32) x (1/2) / P(X = 8, Y = 2)

Similarly, P(1 | X = 8, Y = 2) = (1/32) x (7/32) x (1/2) / P(X = 8, Y = 2)

Since P(0 | X = 8, Y = 2) = P(1 | X = 8, Y = 2), Naïve Bayes will not be able to conclusively provide a class label with sufficient confidence.

1. Naïve Bayes would have no difficulty in classifying the data instance: (X = 3, Y = 5).

P(0 | X = 3, Y = 5) = (3/32) x (5/32) x (1/2) / P(X = 3, Y = 5) P(1 | X = 3, Y = 5) = (5/32) x (3/32) x (1/2) / P(X = 3, Y = 5)

Since P(0 | X = 3, Y = 5) = P(1 | X = 3, Y = 5), Naïve Bayes will not be able to conclusively provide a class label with sufficient confidence.

1. Naïve Bayes would have no difficulty in classifying the data instance: (X = 4, Y = 5).

P(0 | X = 4, Y = 5) = (4/32) x (5/32) x (1/2) / P(X = 4, Y = 5) P(1 | X = 4, Y = 5) = (4/32) x (3/32) x (1/2) / P(X = 4, Y = 5)

Since P(0 | X = 4, Y = 5) > P(1 | X = 4, Y = 5) and the actual class label is 0, Naïve Bayes will have no difficulty in classifying the data instance: (X = 4, Y = 5).

1. Naïve Bayes would have no difficulty in classifying the data instance: (X = 3, Y = 6).

P(0 | X = 3, Y = 6) = (6/32) x (3/32) x (1/2) / P(X = 3, Y = 6) P(1 | X = 3, Y = 6) = (2/32) x (5/32) x (1/2) / P(X = 3, Y = 6)

Since P(0 | X = 3, Y = 6) > P(1 | X = 3, Y = 6) and the actual class label is 0, Naïve Bayes will have no difficulty in classifying the data instance: (X = 3, Y = 6).

1. Consider a two-dimensional dataset for classification, represented as a table below. X and Y are the two attributes, and every cell in the table corresponds to a data instance. Data instances with a value in the cell (0 or 1) represent training instances with their corresponding class labels (0 or 1), while data instances with no value in the cell represent test instances. For example, the data instance: (X = 2, Y = 1) belongs to class 1. There are an equal number of 0s and 1s in the dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Y = 5 | 0 | 0 | 0 |  |
| Y = 4 | 0 | 0 |  | 1 |
| Y = 3 | 0 |  |  | 1 |
| Y = 2 | 0 |  | 1 | 1 |
| Y = 1 |  | 1 | 1 | 1 |
|  | X = 1 | X = 2 | X = 3 | X = 4 |

Two-dimensional data set to consider for classification

1. Compute the following conditional probabilities in the table below using the naïve Bayes approach.

|  |  |
| --- | --- |
|  | **Conditional Probability** |
| P(X = 4 | Class = 1) | 4/7 |
| P(X = 4 | Class = 0) | 0/7 |
| P(Y = 5 | Class = 1) | 0/7 |
| P(Y = 5 | Class = 0) | 3/7 |
| P(X = 4, Y = 5 | Class = 1) | 0 |
| P(X = 4, Y = 5 | Class = 0) | 0 |

1. Using the Naïve Bayes approach, what class label will you predict at (X = 4, Y = 5)? If you faced any difficulty in arriving at your answer, which technique did you use to overcome it?

Answer: We can see that the class conditional probabilities for both classes (0 and 1) are equal to 0 for (X = 4, Y = 5), resulting in undefined posterior probabilities. This issue can be attributed to the brittleness of simple fractions in representing probability values, especially when the number of training instances is small for every combination of attribute values. One way to address this issue is to use correction methods for probability computation, such as the Laplace estimate. See page 224, Equation 4.20.

|  |  |
| --- | --- |
|  | **Laplace estimate of Conditional Probability** |
| P(X = 4 | Class = 1) | 5/11 |
| P(X = 4 | Class = 0) | 1/11 |
| P(Y = 5 | Class = 1) | 1/12 |
| P(Y = 5 | Class = 0) | 4/12 |
| P(X = 4, Y = 5 | Class = 1) | 5/(11X12) |
| P(X = 4, Y = 5 | Class = 0) | 4/(11X12) |

Prior probabilities of Class 0 and 1 can be computed as P(Class = 0) = P(Class = 1) = ½

Since P(X = 4, Y = 5 | Class = 1) X P(Class = 1) > P(X = 4, Y = 5 | Class = 0) X P(Class = 0), we will predict Class = 1 for (X = 4, Y = 5).

1. Consider the data shown in the table below, where A and B are the two attributes, and the class label is either positive (+) or negative (-):

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | Number of (+) records | Number of (-) records |
| a0 | b0 | 80 | 10 |
| a0 | b1 | 30 | 50 |
| a1 | b0 | 48 | 4 |
| a1 | b1 | 32 | 16 |
| a2 | b1 | 10 | 20 |

Use Bayes theorem to answer the following: (Show all steps. Assume conditional independence.)

1. What would be the class label for an instance for which we do not have information about the attributes?

P(+) = 200/300 = 2/3. P(-) = 100/300 = 1/3. The class label would be +.

1. What is the class label of an instance if it is known that A = a2?

P(A = a2 | +) = 10/200 = 1/20.

P(+ | A = a2) = P(A = a2 | +) · P(+) / P(A = a2)

= (1/20) · (2/3) / P(A = a2) = (1/30) / P(A = a2). P(A = a2 | -) = 20/100 = 1/5.

P(- | A = a2) = P(A = a2 | -) · P(-) / P(A = a2)

= (1/5) · (1/3) / P(A = a2) = (1/15) / P(A = a2).

The class label would be -.

1. What is the class label of an instance if it is known that A = a2 and B = b0?

P(+ | A = a2, B = b0) = P(A = a2 | +) · P(B = b0 | +) · P(+) / P(A = a2, B = b0)

= (1/30) · (148/200) / P(A = a2, B = b0) = 0.024 / P(A = a2, B = b0).

P(- | A = a2, B = b0) = P(A = a2 | -) · P(B = b0 | -) · P(-) / P(A = a2, B = b0)

= (1/15) · (14/100) / P(A = a2, B = b0) = 0.009 / P(A = a2, B = b0).

The class label is +.

1. Answer the following questions based on the Bayes Theorem:

(a) Suppose the fraction of undergraduate students who smoke is 15%, and the fraction of graduate students who smoke is 25%. If one-fifth of the college students are graduate students and the rest are undergraduates, what is the probability that a student who smokes is a graduate student?

Given **P**(**S**|**UG**) = 0**.**15, **P**(**S**|**G**) = 0**.**25, **P**(**G**) = 0**.**2, **P**(**UG**) = 0**.**8. We want to compute **P**(**G**|**S**).

According to Bayes’ Theorem,

**P**(**G**|**S**) = (0**.**25 x 0**.**2) / (0**.**15 x 0**.**8 + 0**.**25 x 0**.**2) = 0**.**2941**.** (1)

(b) Given the information in part (a), is it more likely for a randomly chosen college student to be a graduate student or an undergraduate student?

An undergraduate student, because **P**(**UG**) **> P**(**G**).

(c) Repeat part (b) assuming that the randomly chosen student is a smoker.

From (a), **P**(**G**|**S**) = 0**.**2941, and thus **P**(**UG**|**S**) = 1−**P**(**G**|**S**) = 0**.**7059. Thus, a smoker student is more likely to be an undergraduate since **P**(**UG**|**S**) **> P**(**G**|**S**).

(d) Suppose 40% of the graduate students live in a dorm, but only 10% of the undergraduate students live in a dorm. If a student smokes and lives in the dorm, is he or she more likely to be a graduate or undergraduate student? You can assume independence between students who live in a dorm and students who smoke.

First, we need to estimate all the probabilities.

**P**(**D**|**UG**) = 0**.**1, **P**(**D**|**G**) = 0**.**4.

**P**(**D, S**|**G**) = **P**(**D**|**G**) x **P**(**S**|**G**) = 0**.**4 x 0**.**25 = 0**.**1 (using independence between students who live in a dorm and those who smoke.)

**P**(**D, S**|**UG**) = **P**(**D**|**UG**) x **P**(**S**|**UG**) = 0**.**1 x 0**.**15 = 0**.**015.

We need to compute **P**(**G**|**DS**) and **P**(**UG**|**DS**).

**P**(**G**|**D, S**) = (**P**(**D, S**|**G**)**P**(**G**)) / **P**(**D, S**) = (0**.**1 x 0**.**2) / **P**(**D, S**) = 0**.**02 / **P**(**D, S**) (2)

**P**(**UG**|**D, S**) = (**P**(**D, S**|**UG**)**P**(**G**)) / **P**(**D, S**) = (0**.**015 x 0**.**8) / **P**(**D, S**) = 0**.**012 / **P**(**D, S**) (3)

Since **P**(**G**|**D, S**) **> P**(**UG**|**D, S**), he/she is more likely to be a graduate student.

1. The table below shows data collected on a runner’s decision to go for a run or not go for a run, depending on the weather conditions that day. We will use Naïve Bayes (NB) classifier to answer several questions related to this dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Outlook** | **Temperature** | **Humidity** | **Run** |
| Sunny | Hot | High | No |
| Overcast | Cool | Normal | No |
| Sunny | Mild | High | No |
| Overcast | Mild | High | No |
| Sunny | Hot | High | Yes |
| Rainy | Hot | High | Yes |
| Rainy | Mild | High | Yes |
| Rainy | Cool | Normal | Yes |
| Rainy | Cool | Normal | Yes |
| Sunny | Cool | Normal | Yes |
| Rainy | Mild | Normal | Yes |
| Sunny | Mild | Normal | Yes |
| Rainy | Mild | High | Yes |
| Rainy | Hot | Normal | Yes |

|  |  |  |
| --- | --- | --- |
| (Sunny|No)= 2/4 | Hot|No = 1/4 | High|No=3/4 |
| (Sunny|Yes)=3/10 | Hot|Yes = 3/10 | High|Yes = 4/10 |
| (Overcast|No)=2/4 | Cool|No = 1/4 | Normal|No = ¼ |
| (Overcast|Yes)=0/10 | Cool|Yes = 3/10 | Normal|Yes = 6/10 |
| (Rainy|No) =0/4 | Mild|No = 2/4 |  |
| Rainy|Yes = 7/10 | Mild|Yes = 4/10 |  |

1. Given the data in the table above, is a person more likely to go for a run or not? Justify your answer.

P(Run = Yes) = 10/14 > P(Run = No) = 4/14

A person is more likely to run given no information about the weather conditions.

1. How would Naïve Bayes classify an unseen data point X = {Sunny, Mild, Normal}? Show your work.

P(Yes|X) = (P(X | Yes) P(Yes)) / P(X)

Assuming attributes are conditionally independent given class,

P(X | Yes) P(Yes) = P(Sunny|Yes)P(Mild|Yes)P(Normal|Yes)P(Yes)

= (3/10)(4/10)(6/10)(10/14) = 9/175 = 0.0514

Similarly,

P(No|X) = (P(X | No) P(No)) / P(X)

P(X | No) P(No)= P(Sunny|No)P(Mild|No)P(Normal|No)P(No)

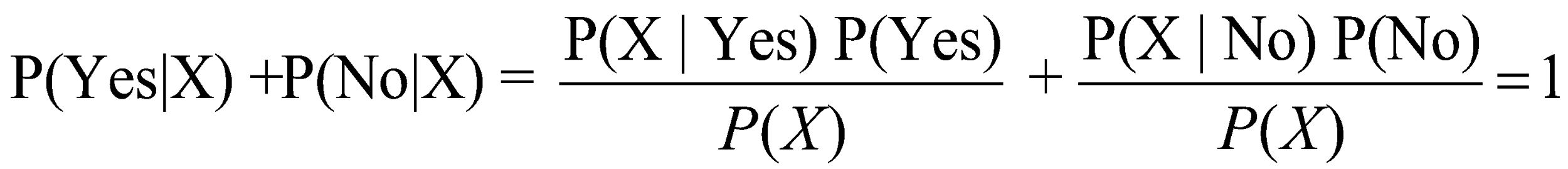
=(2/4)(2/4)(1/4)(4/14) = 1/56 = 0.0179

P(Yes|X) = 0.0514/P(X) > 0.0179/P(X) = P(No|X)

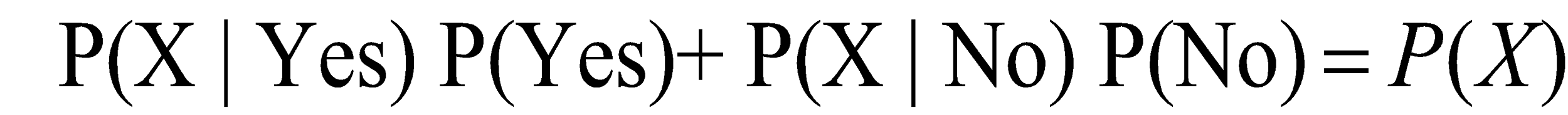
Therefore, X would be classified as Run = Yes.

**IMPORTANT NOTE:** Some students attempted to compute the exact value of P(X) either directly from the table (counting the total occurrences of X) and assigned the class label to be “Yes” if P(Yes | X) > 0.5 and otherwise assigned the class label to be “No.” Note that the probability calculated directly from the table is not consistent with the Naïve Bayes approach.

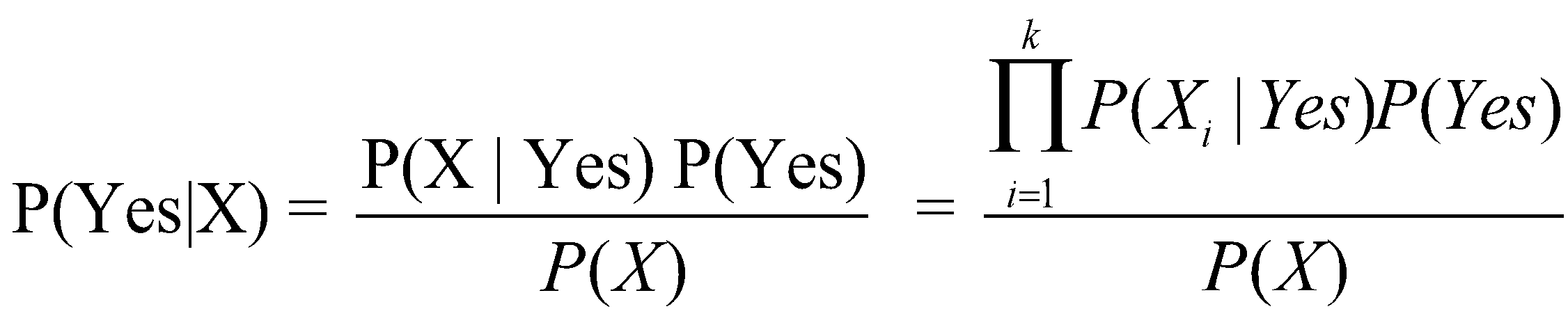
To understand this further, we have



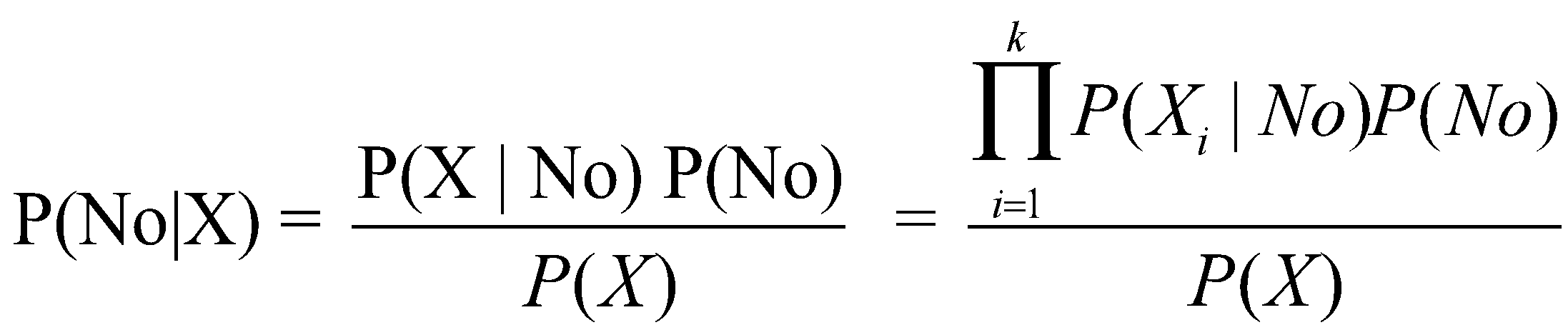
which implies

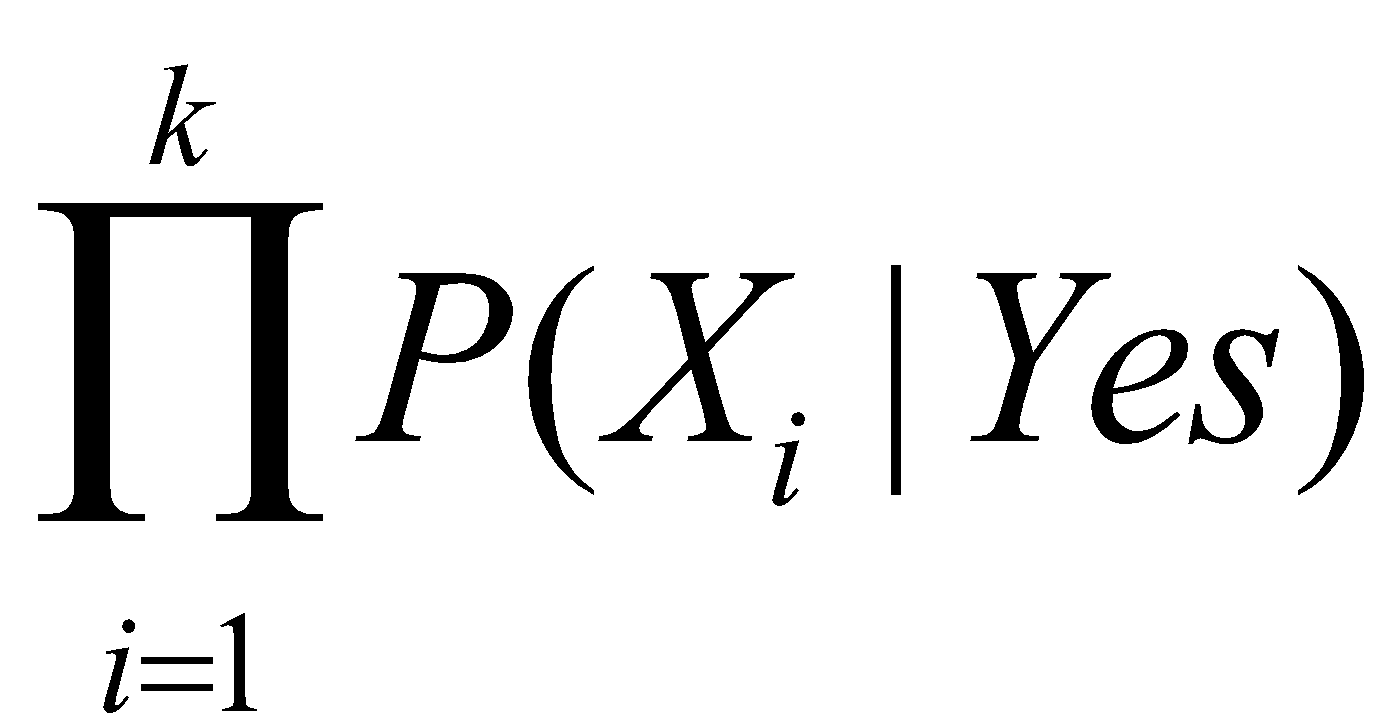
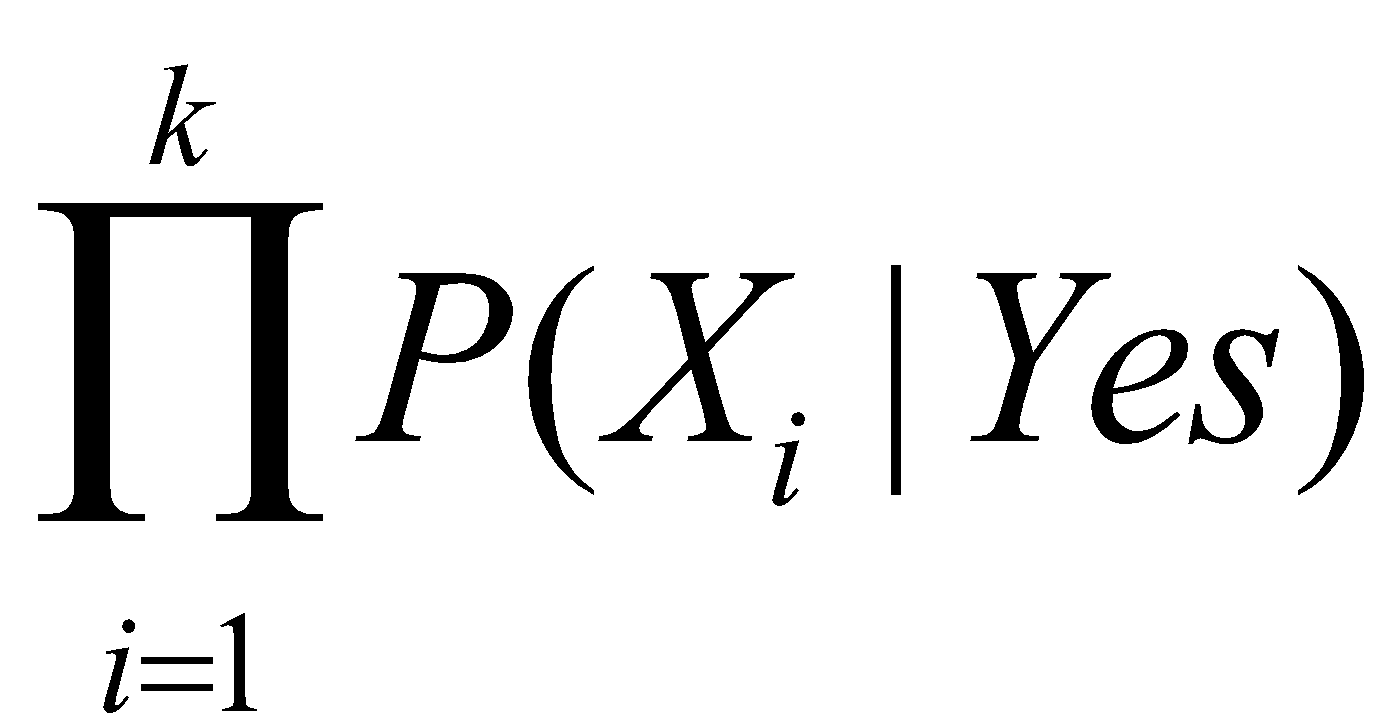
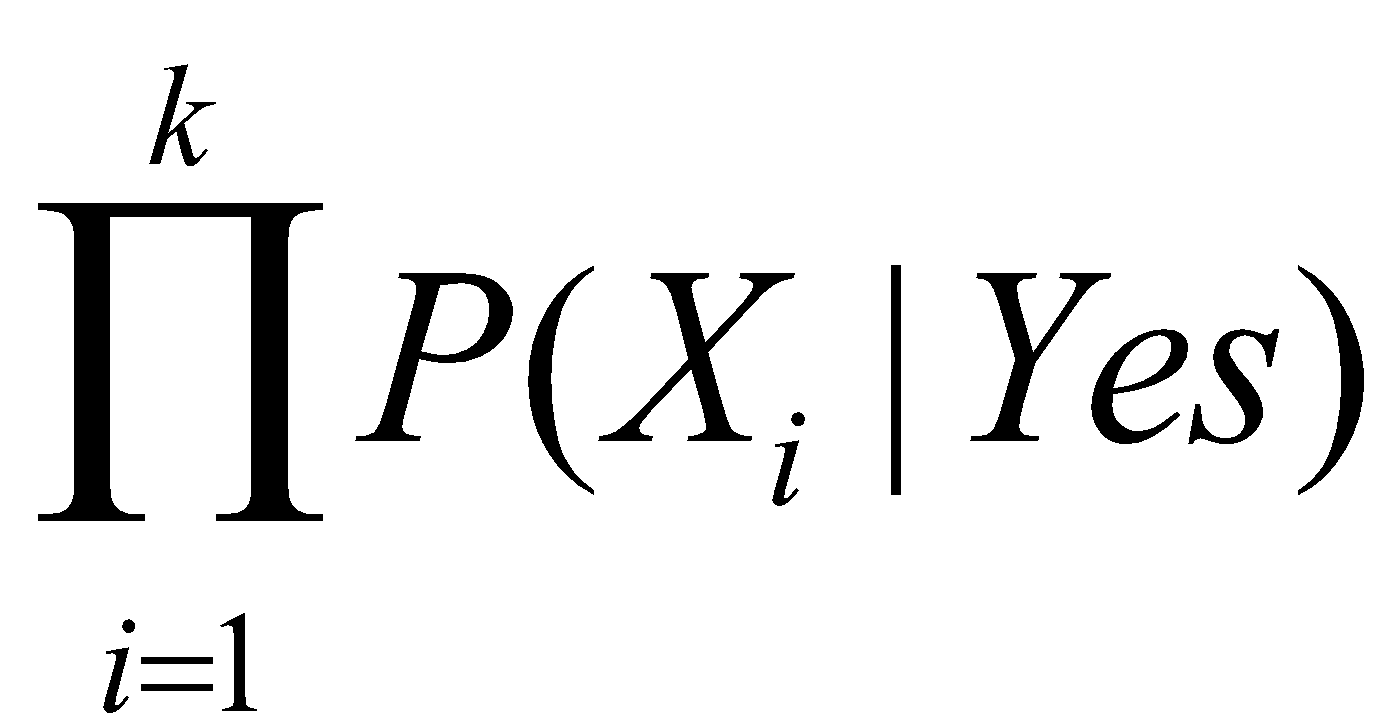
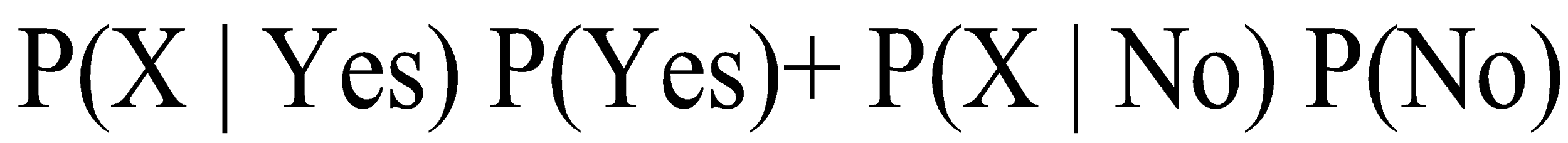


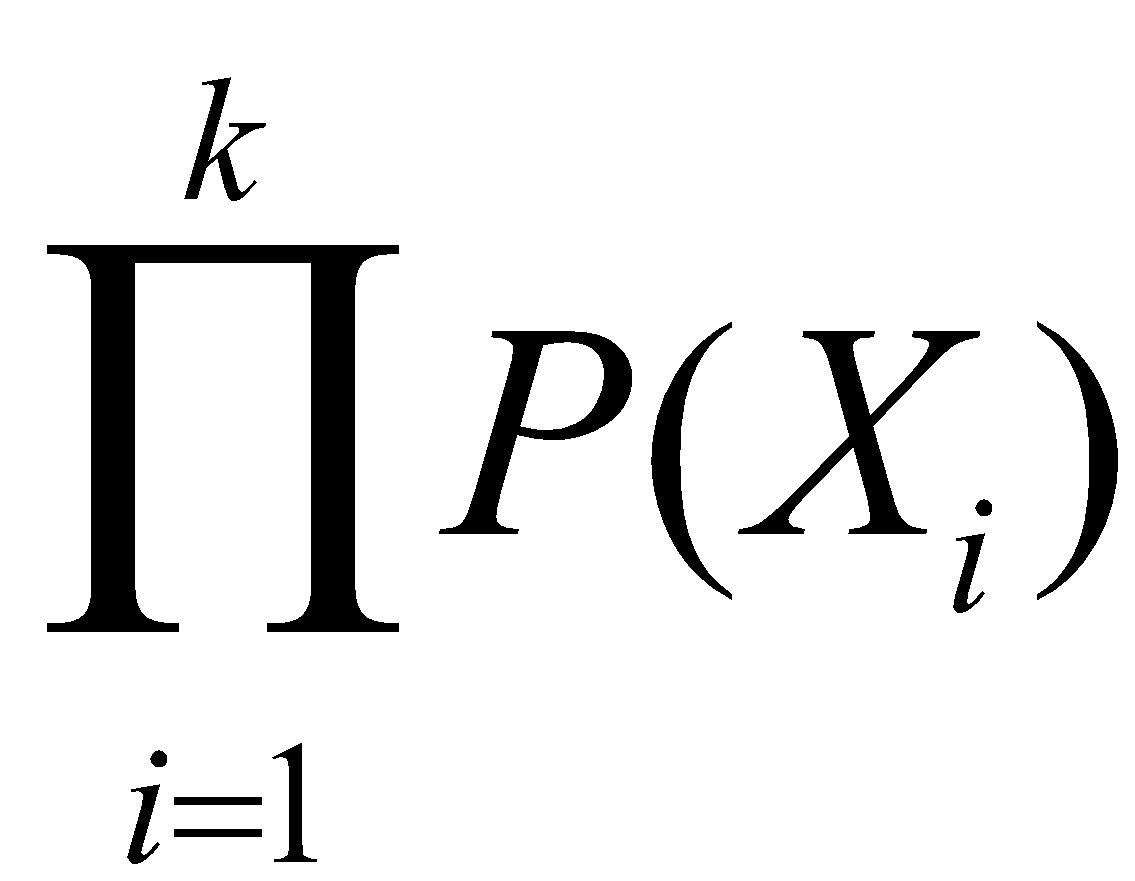
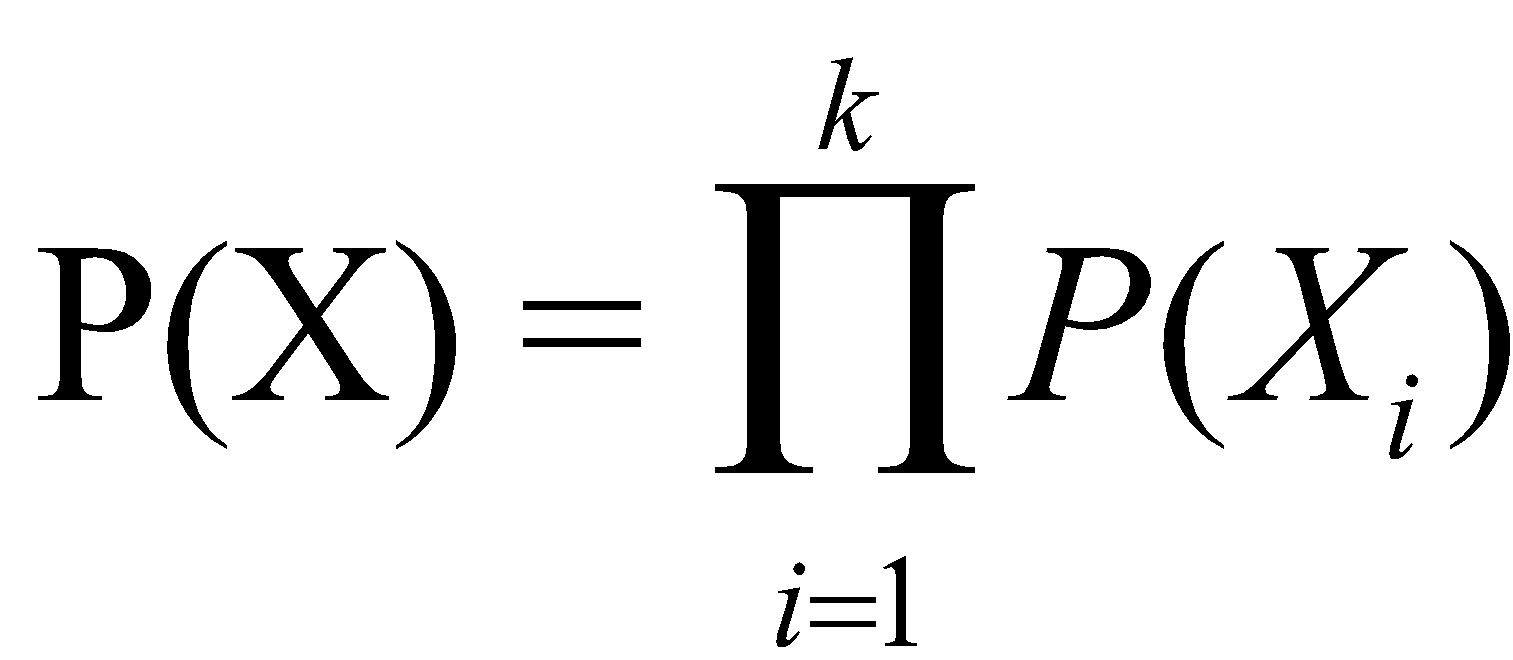
Note that, while computing P(Yes | X) and P(No | X) as



and



respectively, one should remember that the term  is the true value of P(X|Yes) if and only if the above assumption of class conditional independence among the attributes *exactly* holds in the data. Otherwise, the terms  and  are just the estimated values of P(X|Yes) and P(X|No) respectively in which case the value of P(X) computed directly from the data may not be equal to , and consequently, P(Yes|X) and P(No | X) would not sum up to 1. Therefore, the correct procedure is to leave the denominator as it is and directly compute the numerators for both P(Yes | X) and P(No | X) and compare them to decide the final class label of X.

Besides that, some students computed P(X) as which is again technically incorrect. Note that the classifier Naïve Bayes assumes the class conditional independence among the attributes, NOT independence among the attributes, whereas requires the latter to be true.

1. Assume that the only information you have about the weather outside is that temperature is mild. What is NB’s prediction of whether a person will run or not? Show your work.

For ease of notation, let X denote event {Temperature = Mild}. Then, P(Yes|X) = P(X | Yes) P(Yes) / P(X)

P(X | Yes) P(Yes) = 4/14 = 0.2857

P(No|X) = P(X | No) P(No) / P(X)

= 2/14 = 0.1428

P(Yes|X) = 0.2857/ P(X) > 0.1428/ P(X) = P(No|X)

Therefore, given *Temperature* = ‘Mild,’ a person is more likely to run that day.

1. In addition to knowing that temperature is mild that day, you also know that humidity is high. What is NB’s prediction of whether a person will go for a run or not?

For ease of notation, let

X denote event {Temperature = Mild, Humidity = High},   
X1 denote event {Temperature = Mild}, and   
X2 denote event {Humidity = High}.

Then, P(Yes|X) = P(X | Yes) P(Yes) / P(X)

Assuming class conditional independence between attributes given the class,

P(X| Yes) P(Yes) = P(X1|Yes)P(X2|Yes)P(Yes) = (4/10)(4/10)(10/14)= 4/35 = 0.114

Similarly,

P(No | X) = P(X| No) P(No) / P(X)

Assuming class conditional independence between attributes,

P(X | No) P(No) = P(X1|No)P(X2|No)P(No) = (2/4)(3/4)(4/14)= 6/56 = 0.107

P(Yes | X) = 0.114/ P(X) > 0.107/ P(X) = P(No | X)

Therefore, the given temperature is mild and humidity is high, a person is likely to go for a run.

1. Now let us go back and compute prediction for a complete data point. In addition to knowing that the temperature is mild and the humidity is high, assume you also know that the outlook is overcast. Is a person more likely to go for a run?

For ease of notation let

X denote event { Outlook = Overcast, Temperature = Mild, Humidity = High}   
X1 denote event{Outlook = Overcast}   
X2 denote event{Temperature = Mild}   
X3 denote event { Humidity = High}

P(Yes| X) = P(X | Yes) P(Yes) / P(X)

Assuming class conditional independence between attributes,

P(X|Yes)P(Yes) = P(X1|Yes) P(X2|Yes) P(X3|Yes)P(Yes) = (0)(4/10)(4/10)(10/14) = 0

P(No| X) = P(X | No) P(No) / P(X)

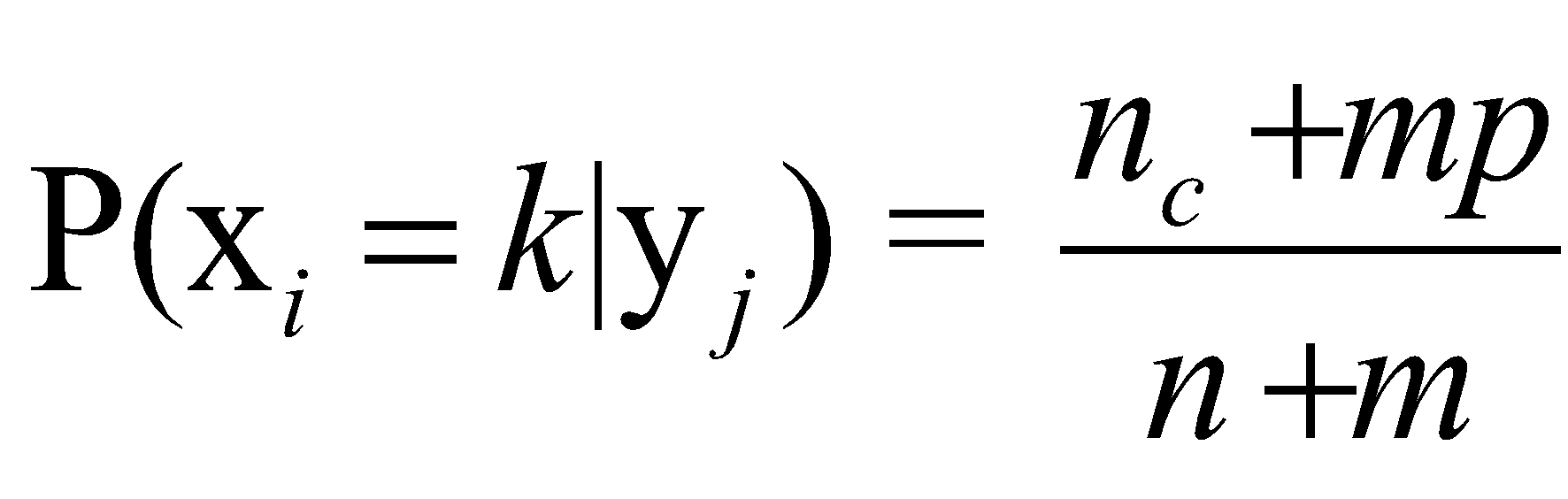
Assuming class conditional independence between attributes,

P(X|No)P(No) = P(X1|No) P(X2|No) P(X3|No)P(No) = (2/4)(2/4)(3/4)(4/14) = 3/56. P(No|X) = 3/(56\*P(X)) > 0 = P(Yes|X)

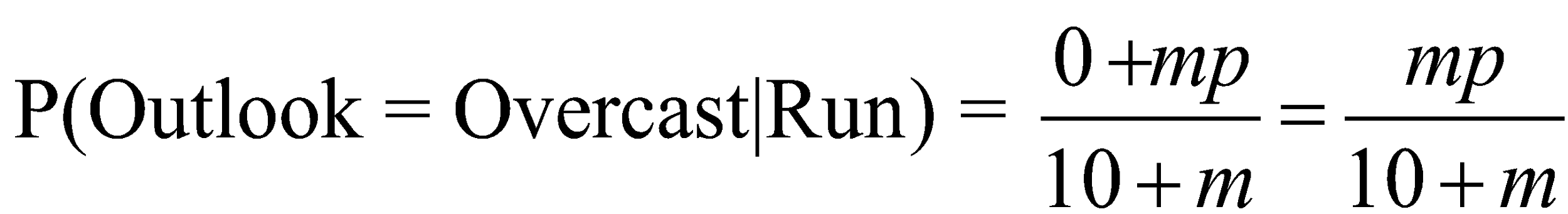
Since P(No | X) > P(Yes | X), therefore according to Naïve Bayes, a person is not likely to go for a run given { Outlook = Overcast, Temperature = Mild, Humidity = High}. However, although P(No | X) > P(Yes | X), P(Yes | X) = 0. Therefore, it is not completely clear whether a person is likely to go for a run that day.

1. What went wrong in e)? What approach would you use to fix it? Explain your answer.

In the previous case, although P(No | X) > P(Yes | X), but P(Yes | X) = 0 since no record was observed with Outlook = Overcast and class label = *Run*. It could be possible that the event {Outlook = Overcast} is extremely rare due to which it was never observed in class *Run* and as a result, the overall posterior probability for this class turned out to be zero. Therefore, it is not completely clear whether a person is likely to go for a run that day. This can be fixed with the use of m-estimate approach in which p(xi|yj), the probability of attribute xi given class label yj is computed as



where n is the total number of records in the training data, nc is the number of records in training data for which the attribute xi takes the value k, m is the equivalent sample size (constant hyper-parameter), and p is the prior probability of observing the attribute xi to be equal to k among records with class label yj. Using this approach, P(Overcast|Yes) can be estimated as



1. The table below shows data collected on a runner’s decision to go for a run or not go for a run, depending on the weather conditions that day. We will use Naïve Bayes (NB) classifier to answer several questions related to this dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Outlook** | **Temperature** | **Humidity** | **Run** |
| Sunny | Hot | High | No |
| Overcast | Cool | Normal | No |
| Sunny | Mild | High | No |
| Overcast | Mild | High | No |
| Sunny | Hot | High | Yes |
| Rainy | Hot | High | Yes |
| Rainy | Mild | High | Yes |
| Rainy | Cool | Normal | Yes |
| Rainy | Cool | Normal | Yes |
| Sunny | Cool | Normal | Yes |
| Rainy | Mild | Normal | Yes |
| Sunny | Mild | Normal | Yes |
| Rainy | Mild | High | Yes |
| Rainy | Hot | Normal | Yes |

|  |  |  |
| --- | --- | --- |
| (Sunny|No)= 2/4 | Hot|No = 1/4 | High|No=3/4 |
| (Sunny|Yes)=3/10 | Hot|Yes = 3/10 | High|Yes = 4/10 |
| (Overcast|No)=2/4 | Cool|No = 1/4 | Normal|No = ¼ |
| (Overcast|Yes)=0/10 | Cool|Yes = 3/10 | Normal|Yes = 6/10 |
| (Rainy|No) =0/4 | Mild|No = 2/4 |  |
| Rainy|Yes = 7/10 | Mild|Yes = 4/10 |  |

1. Given the data in the table above, is a person more likely to go for a run or not? Justify your answer.

P(Run = Yes) = 10/14 > P(Run = No) = 4/14

A person is more likely to run given no information about the weather conditions.

1. Assume that the only information you have about the weather outside is that temperature is mild. What is NB’s prediction of whether a person will run or not? Show your work.

P(Temperature = Mild | Yes) P(Yes) =4/10\*10/14= 4/14

P(Temperature = Mild | No) P(No) = 2/4\*4/14=2/14.

Given Temperature = ‘Mild,’ a person is likely to run that day.

1. In addition to knowing that temperature is mild that day, you also know that humidity is high. What is NB’s prediction of whether a person will go for a run or not?

P(Temperature = Mild, Humidity = High | Yes) P(Yes) =4/10\*10/14\*4/10= 0.114

P(Temperature = Mild, Humidity = High | No) P(No) = = 2/4\*4/14\*3/4=0.107

Given temperature is mild and humidity is high, a person is likely to go for a run.

1. Now let us go back and compute prediction for a complete data point. In addition to knowing that the temperature is mild and humidity is high, assume you also know that the outlook is overcast. Is a person more likely to go for a run or not.

P(Outlook = Overcast, Temperature = Mild, Humidity = High | Yes) P(Yes)= 0

P(Outlook = Overcast, Temperature = Mild, Humidity = High | No) P(No) = 2/4\* 2/4\*4/14\*3/4=3/56 (0.0536)

A person is not likely to go for a run that day.

1. What went wrong in e)? What approach would you use to fix it? Explain your answer.

Probability of Outlook = Overcast given Run = Yes is zero, which makes the overall posterior probability for this class zero. This can be fixed with the use of m-estimate.

1. How would Naïve Bayes classify an unseen data point X = {Sunny, Hot, Normal}? Show your work. Comment on the behavior of Naïve Bayes in the case of new unseen data.

P(X | Yes) P(Yes) =3/10\*3/10\*6/10\*10/14= 0.0386

P(X | No) P(No) = 2/4\*1/4\*1/4\*4/14=0.00893

X would be classified as Run = Yes.

NB successfully handles unseen data.

1. Naïve Bayes

A screenshot of a cell phone

Description automatically generated

1. Explain how Naïve Bayes performs on the dataset shown in the figure above, where the two target classes are A and B.

Naïve Bayes will not do well on this data set because the conditional probabilities for each distinguishing attribute given the class are the same for both class A and class B.

1. If each class is further divided, such that there are four classes (A1, A2, B1, and B2), will Naïve Bayes perform better?

The performance of Naïve Bayes will improve considerably on the subclasses because the product of conditional probabilities for the distinguishing attributes given the subclass will be different for each subclass.

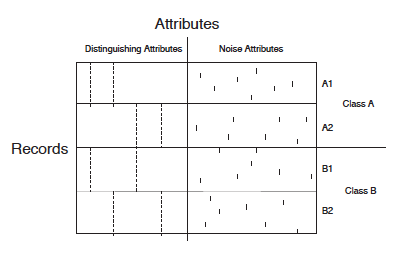
1. How will a decision tree perform on this data set (for the two-class problem)? What if there are four classes?

For the two-class problem, a decision tree may not perform well (especially if there are a large number of noise attributes) because the Information Gain for each distinguishing attribute would be similar to the Information Gain for the Noise attributes, and hence the noise attributes would have an equal chance of being chosen for splitting. If there are four subclasses, then the decision tree will improve considerably, because the distinguishing attributes will have higher Information Gain as compared to the noise attributes, and hence will be chosen for splitting. A decision tree with perfect classification can be obtained within 2 levels of splitting.

1. You are given the choice of the following classifiers: Naïve Bayes, k Nearest Neighbor, Linear SVM and Decision Trees. For each dataset in the left column, indicate the best and the worst choice. Provide a brief explanation. If you think there are more than one best or worst, just choose and provide an explanation for one of them.

|  |  |
| --- | --- |
| Figure | Answer |
|  | Best Choice: kNN  Worst Choice: Linear SVM or Naïve Bayes  Explanation:  kNN: kNN can separate classes with arbitrary boundaries, kNN classify together instances that are in close proximity  Linear SVM: Linear SVM can only work on a dataset with a linear boundary between the classes. Naïve Bayes: NB will also not work well for this data set due to attribute dependency.  Decision Tree can only draw rectilinear split at each split, hence will be too complex for this figure. |
| There are a large number of noise attributes, and they are randomly filled with 1s and 0s | Best Choice: Naïve Bayes or Linear SVM Worst Choice: kNN, DT  Explanation:  Naïve Bayes: NB will do very well in this data set because each discriminating attribute has higher conditional probability in one class over the other, and the overall classification is done by multiplying these individual conditional probabilities.  Linear SVM: Is robust to noisy attributes, especially if we introduce slack variables.  kNN’s proximity measure will be impaired because it will bias too much to the noisy attributes. DT suffers with the presence of irrelevant attributes. |

1. Consider the two-class data set shown in the figure below. The dataset consists of 4 distinguishing attributes and many more noise attributes. All attributes are binary. A vertical dash represents a ‘1’ for the record on that specific attribute and the absence of a vertical dash represents a 0.



1. Is there a single attribute in the dataset that can be used to distinguish between Class-A and Class-B?

No

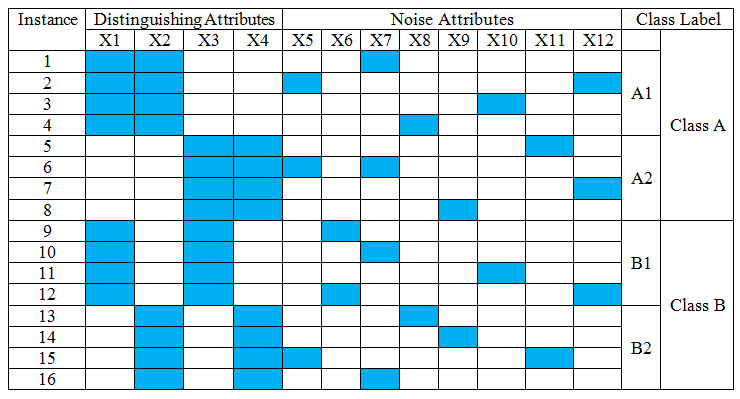
1. Comment on the performance of Decision Trees on this dataset.

The decision tree can still perform well as long as there not too many noise attributes. If there are too many noise attributes, then they are likely to be selected at many internal nodes and thus make the model too complex (e.g., see Figure 3.27 in Tan et al. Second edition).

1. Comment on the performance of naive Bayes classifier on this dataset.

Bad

1. Naïve Bayes.



(a) Explain how naive Bayes performs on the data set above when we only consider two classes: A and B.

NB will not do well on this data set because the conditional probabilities for each distinguishing attribute given the class are the same for both class A and class B.

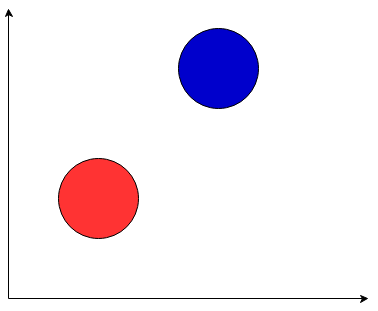
(b) If each class is further divided such that there are four classes (**A**1, **A**2, **B**1, and **B**2), will naive Bayes perform better?

The performance of NB will improve on the subclasses because the product of conditional probabilities among the distinguishing attributes will be different for each subclass.

(c) How will a decision tree perform on this data set (for the two-class problem)? What if there are four classes?

For the two-class problem, the decision tree will not perform well because the entropy will not improve after splitting the data using the distinguishing attributes. If there are four classes, then the decision tree will improve considerably, because the distinguishing attributes will have better values for measures such as entropy, as compared to the noise attributes.

1. For any two-dimensional dataset, we can use visual inspection to determine whether Naïve Bayes can correctly estimate the class labels at different points, thereby assessing the suitability of naïve Bayes for classification on the given dataset. Determine whether the Naïve Bayes classifier would be suitable for each of the datasets shown in the figures below. The two axes in each of the figures represent the two attributes, and the target binary classes are red and blue. The red and blue patches are of the equal density of points and there are an equal number of red and blue points in all subfigures, and hence the prior probabilities of both classes is 0.5.

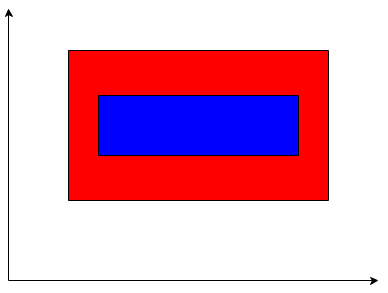
A)

Answer: Yes. The conditional probability of red class is higher in the red patch along both axes. Similarly, the conditional probability of blue class is higher in the blue patch along both axes.

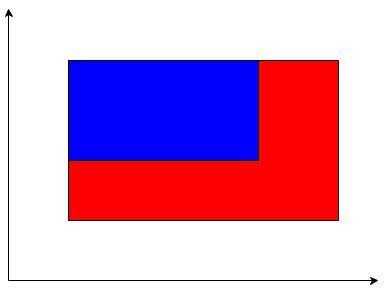
B) A picture containing drawing

Description automatically generated

Answer: No. For the leftmost red patch, the conditional probability of blue class along the X-axis is 0, and hence it will be correctly classified as red. Similarly, for the rightmost blue patch, the conditional probability of red class along the X-axis is 0, and hence it would be correctly classified as blue. However, for the middle red and blue patches, the conditional probability of red and blue classes are equal along both axes. As a result, the middle red and blue patches would be misclassified.

C)

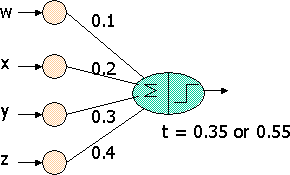
Answer: Yes. In the blue patch, the conditional probability of blue class is higher than red class along the Y-axis and equal along the X-axis. As a result, it will be correctly classified as blue. For the red patches surrounding the blue patch, the conditional probability of the blue class is 0 either along the X-axis or the Y-axis (or both). Hence, the red patches will be correctly classified as red.

D) 

Answer: Yes. In the blue patch, the conditional probability of blue class is higher than red class along both axes and hence will be correctly classified as blue. For the red patches surrounding the blue patch, the conditional probability of the blue class is 0 either along the X-axis or the Y-axis (or both), and hence will be correctly classified as red.

# **Artificial Neural Networks**

1. For the perceptron below:



What inputs will make the output 1 if the threshold t is 0.35, but NOT if the threshold is 0.55? After writing a Boolean expression in terms of w, x, y, and z for the set of possible inputs, identify one of those inputs in the lists below.

a) w=0; x=0; y=1; z=0 b) w=0; x=1; y=0; z=0   
c) w=0; x=1; y=1; z=0 d) w=0; x=1; y=0; z=1   
e) w=1; x=1; y=0; z=1 f) w=1; x=1; y=1; z=0

If z=1, the lower threshold is met, so we must make sure the sum of inputs that are 1 does not go above 0.55. The sum will be at least 0.6 if either x or y is 1, but it doesn't matter

whether w=1. Thus, one term in the correct Boolean function is x'y'z (we use primes to indicate negation). If z=0, then to get above 0.35 threshold, y must be 1, and at least one of w and x must

also be 1. However, if w=x=y=1, then the sum is 0.6, which is too high. In the last case, where z=y=0, it is not possible to reach the 0.35 threshold. Thus, the Boolean function is x'y'z + w'xyz' + wx'yz'

For the inputs in the list, the sums would be:

a) 0.3 b) 0.2 c) 0.5 d) 0.6 e) 0.7 f) 0.6

So the outputs would be:

a) 0 b) 0 c) 1 d) 0 e) 0 f) 0

1. should be selected

# **Deep Learning**

1. Select which of the following methods can avoid the vanishing gradient problem in learning deep neural networks? (You can select multiple choices as your answer if necessary.) No explanation needed.
2. Use of larger training sets.
3. Use of cross-entropy loss function instead of the sum of squared errors, when activation function at the output layer is sigmoid.
4. Use of deeper ANN architectures with a larger number of hidden layers.
5. Replacing sigmoid activation function with Rectified Linear Units (ReLU) at the hidden layers

Answer: b and d.

1. State whether the following statements are True or False, giving a one sentence justification.
2. In the back-propagation algorithm for training ANN models, the gradients of weights at the k+1th layer can be computed using the gradients of weights at the kth layer.

Answer: False. The gradients are computed starting from the output layer to the hidden layers, rather than the opposite direction.

1. While applying an ANN model on a test instance, the activations at nodes at the k+1th layer can be computed using the activations at nodes at the kth layer.

Answer: True. The activations at kth layer can be directly computed using the activations at k+1th layer as inputs.

1. An ANN training procedure is said to suffer from the vanishing gradient problem if the training errors vanish to zero while the test errors are still large.

Answer: False. The vanishing gradient problem results in zero gradients of training loss with respect to weights at the hidden and output layers. This does not mean that the training errors are 0 while test errors are large, which is closer to the phenomena of overfitting.

1. If at a given iteration of the back-propagation algorithm, the ANN model perfectly classifies all training instances, then the gradients of loss with respect to weights at all layers will be 0.

Answer: True. If the training loss is 0, then the gradient at the output node is 0, and hence the gradients at all hidden nodes are 0.

# **Support Vector Machine**

* + - 1. Suppose you are given a data set consisting of nominal attributes, such as color, which takes values such as red, blue, green, etc. Can you use this data set directly to train an SVM? If not, how will you transform these attributes into a representation that can be used to train an SVM?

Solution:

No, since SVM works naturally only with numerical valued data. So, the above data set can be used with an SVM by creating a binary attribute for each nominal attribute-value pair.

Alternative Solution: One can define a kernel for nominal attributes and then use it within an SVM.

Similarity: Both are ensemble methods, i.e., they combine the predictions from multiple classification models.

* + - 1. For a linear support vector machine (SVM) classifier with as the decision boundary, a training record is termed a support vector if it satisfies the following equality:

, where .

The figure below shows a linear SVM classifier trained over a two-dimensional data set A, consisting of five positive points, P1 to P5, and five negative points, N1 to N5.

Figure. Classification Data set A

w.x+b = 0

w•x+b = +1

w•x+b = -1

P1

P2

P3

P4

P5

N1

1

N2

1

N3

1

N4

1

N5

1

1. Identify all the support vectors of this SVM classifier, either by circling them in Figure above or by listing them below.

Answer: Support vectors: P4, P5, N1, N3

1. Suppose that we add 5 positive points and 5 negative points to data set A to obtain a new data set B as shown in Figure below. (Note that the positions of points P1 to P5 and N1 to N5 have not been changed from data set A to data set B.)

P1

P2

P3

P4

P5

N1

1

N2

1

N3

1

N4

1

N5

1

If we train a linear SVM on data set B using the same hyper-parameter settings as that for A, which support vectors (if any) do you expect to be different from your answer in part (a)? Will this result in a different SVM classifier trained on data set B than the one trained on data set A? Briefly explain.

Answer: The support vectors will not change from data set A to data set B as the newly added points are farther away from the margin hyperplane separating positive and negative points in data set A, compared to the support vectors of data set A. Since the SVM classifier only depends on the support vectors, the SVM classifier trained on data set B will remain the same as the SVM classifier trained on data set A.

1. Instead of SVM, if you used logistic regression classifier for this problem, do you expect to learn a different logistic regression model by training on data set B than by training on data set A? Briefly explain.

Answer: Unlike SVM, logistic regression is not just affected by the support vectors (closest training points to the maximum margin hyperplane) but by every positive and negative point that contributes to the total loss of overall training instances which need to be minimized. Hence, in general, the logistic regression model trained over data set B can be different from the model trained over data set A.

# **Ensemble Methods**

* + - 1. State one similarity and dissimilarity between bagging and boosting.

Similarity: Both are ensemble methods, i.e., they combine the predictions from multiple classification models. Both use sampling with replacement.

Differences: Boosting assigns a weight to each training example depending on the difficulty faced by the current classifier in classifying it, while bagging combines the predictions from classifiers trained on different samples of the training set in an unweighted fashion. Also, boosting assigns different weights to the classifiers based on how well they perform, while bagging assigns the same weight to all classifiers.

* + - 1. Describe 2 ways in which boosting, as described in class, tries to improve on bagging. Also, describe a possible problem with boosting.

Boosting changes the weights with which individual instances are sampled to better handle instances that are being misclassified. Further, boosting assigns more weight to classifiers that have higher accuracy. Bagging doesn’t do either of these things.

Since boosting is focusing more on instances that are being misclassified, it can overfit the data in some situations, e.g., noisy data.

* + - 1. Alan recently came across the concept of ensemble methods in a data mining class and decided to apply it to stock market prediction. In order to predict whether the stock market will rise or fall on a given day, he decided to flip a coin 1000 times and predicted the stock market to go up if heads turn up in majority and vice-versa. He thinks that this approach could get him a better prediction of the stock market because an ensemble of independent classifiers could potentially obtain a better prediction. Do you agree with him? Give a brief justification.

No, because the ensemble of base classifiers gets a better classification performance only when at least some of the base classifiers are better than a random classifier. In this case, all the base classifiers (the coin tosses) are random and therefore, their ensemble would not do any better than any other random classifier.

* + - 1. For a binary classification problem, you are given a collection of base classifiers where every base classifier has an error rate of e. The ensemble prediction is simply the majority vote of the predictions of the base classifiers. State whether the following statements are “true” or “false.” No explanation needed.

1. If e = 0.5and the predictions of all base classifiers are independent, the error rate of the ensemble classifier will be smaller than e.

Answer: False

1. If e = 0.3 and the predictions of all base classifiers are independent, the error rate of the ensemble classifier will be smaller than e.

Answer: True

1. If e = 0.3 and all base classifiers are identical, the error rate of the ensemble classifier will be smaller than e.

Answer: False

* + - 1. On the left, you are given four classification scenarios that are possible for different settings of bias and variance of the classifier. On the right, you are given four types of classifiers. You need to match the classification scenarios to their corresponding classifiers either by drawing arrows or by writing down matching pairs.

**Classification Scenarios** **Classifier Types**

1. Low Bias, Low Variance (i) Underfitting Classifier
2. High Bias, Low Variance (ii) Overfitting Classifier
3. Low Bias, High Variance (iii) Ideal Classifier
4. High Bias, High Variance (iv) Worst Classifier

Answer: a – iii, b – i, c – ii, d – iv.

# **Class Imbalance Problem, Classification Evaluation Measures, and ROC**

1. Consider a test data of 1000 samples with two classes: + class (100 samples) and – class (900 samples). We have two random classifiers C1 and C2. Classifier C1 classifies test data to + class randomly with a probability p and classifier C2 classifies test data to + class randomly with a probability 2p.
2. What is the expected TPR and FPR for C1 and C2?

For C1, TPR=FPR=p. For C2, TPR=FPR=2p

1. Is C2 a better classifier than C1? Hint: The random guess line in an ROC curve corresponds to TPR = FPR.

No, when p takes continuous value in the range [0, 1], both ROC curves of C1 and C2 are the same with the random guess line, which is TPR=FPR.

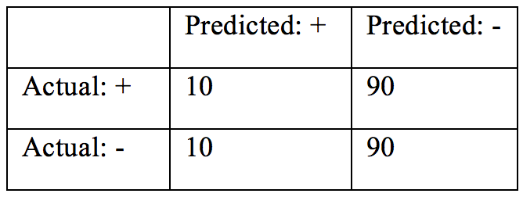
1. The expected precision for both C1 and C2 is 1/10. Expected recall for C2 is twice that of C1 (2p and p, respectively). If we use precision and recall as the evaluation metrics, C2 appears to be a better classifier than C1. Which evaluation metric pair between {TPR and FPR} and {precision and recall} do you think is correctly indicating the relative performance of C2 and C1?

{TPR and FPR}.

Since C1 and C2 are both random classifiers, expected precision for both of them are always the same, so precision contains no information here. Besides, the recall will make C2 a winner, but recall evaluation is just comparing the random probabilities, which is apparently meaningless. In practice, C1 and C2 are equivalent because they are both randomly guessing, so using {TPR and FPR} is more proper for their relative performance.

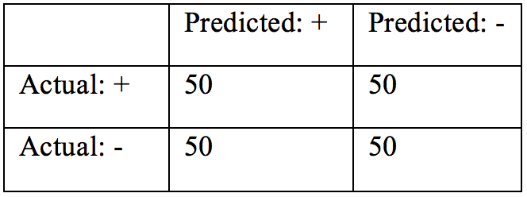
1. Consider a test data of 200 samples with two classes (+ and -). The following are two confusion matrices for Classifier C1 and C2

Confusion matrix for C1



**C1: Precision = 50%; Recall/TPR = 10%; F1-measure = 16.67%; FPR = 10%**

Confusion matrix for C2.



**C2: Precision = 50%; Recall/TPR = 50%; F1-measure = 50%; FPR = 50%**

(i) Based on the information given above, comment on the relative performance of these two classifiers.

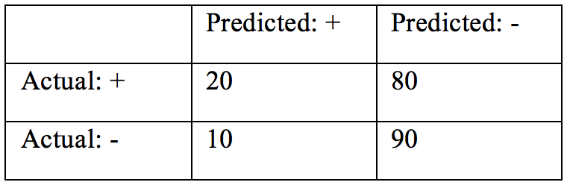
None of C1 and C2 is better than the other. The two classifiers are random, as shown by their TPR/FPR (C1= 0.1/0.1; C2=0.5/0.5), which lies on the random guess line of ROC curve

(ii) Which evaluation metric pair between {TPR and FPR} and {precision, recall, F1-measure} do you think is correctly indicating the relative performance of classifiers C1 and C2? **Explain briefly.**

{TPR, FPR}. Higher F-1 measure might be caused by C2 having a higher probability of assigning an instance to (+) class, which cause more instance to be classified as (+) (but does not mean that those + predictions are accurate)

(iii) Consider a new Classifier, C3, with the following confusion matrix on the same test data

Confusion matrix for C3



**Precision = 66.67%; Recall = 20%; F1-measure = 30.7%; FPR = 10%**

Given the choice of C1, C2 and C3, which classifier would you prefer over the other two? **Explain briefly.**

C3 is the best classifier because it has the highest TPR/FPR=2. TPR / FPR provides a better evaluation of a model across different population distribution. So, if C3 is better than C2 and C1 based on TPR / FPR on this dataset, it will still perform similarly when tested on different population with a different distribution.

1. We have test data of 1000 samples with two classes: a + class (100 samples) and a – class (900 samples). Consider a random classifier C0 that classifies a test data instance to the + class randomly with a probability p.
2. What is the expected precision and recall for C0?

Precision(C0) = 1/10. Recall(C0) = *p*.

1. Write the expression for the F­measure of C0?

F-measure =

1. Consider another classifier C1 whose F­measure is known to be 0.15. Is C1 better than a random classifier?

The F-measure cannot always be an appropriate measure for any classifier under any scenario. So, we can hardly say that C1 is better than a random classifier depending only on the given information.

However, in our case where the recall of the random classifier C0 is a constant p, let

So, when the recall p<0.3, C1 is better than C0.

1. You are given a classifier that attempts to predict whether it will rain tomorrow (+) or not (-). The confusion matrix below gives the results of this classification algorithm on a sample of 1000 consecutive days:

|  |  |  |
| --- | --- | --- |
| actual/ predicted | + | - |
| + | 30 | 20 |
| - | 50 | 900 |

(a) Compute the accuracy, precision, recall, and F measure for the confusion matrix.  
(Compute precision, recall, and the F-measure with respect to + class only.)

Accuracy = 930/1000 = 0.93  
Precision = 30/(30+50) = 0.375  
Recall = 30/(30+20) = 0.6  
F-measure = (2 · 30)/(2 · 30+50+20) = 60/130 = 0.46  
  
(b) Which of these metrics is a poor indicator of the overall performance of your  
algorithm? Which of these metrics is a good indicator of the overall performance? Explain briefly why this is the case?

Accuracy is a poor indicator of overall performance.  
F-measure is a good indicator of overall performance.  
There is a class imbalance problem (the + class is the minority), so accuracy is not a good indicator in this situation, while the F-measure is an appropriate metric.

(c) Construct a trivial rule-based model that gives better accuracy than the classification algorithm above.

{ } → -  
Accuracy = 950/1000 = 0.95

1. You are given the task of predicting whether it will rain tomorrow (+) or not (-) based on a sample of 1000 consecutive days up to today. Given the results of a classification algorithm in the confusion matrix below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted | | |
| Actual |  | + | - |
| + | 20 | 50 |
| - | 80 | 850 |

1. Compute accuracy, precision, recall, and F-measure with respect to ‘+’ class.

Accuracy = 0.87

Precision = 0.2

Recall = 0.28

F-measure = 0.23

1. Which of these metrics is a poor indicator of the overall performance of your algorithm? Which of these metrics is a good indicator of the overall performance? Give a one sentence reason why this is the case?

Accuracy is a poor indicator of the overall performance of the algorithm. In the case of imbalanced class problem, accuracy does not evaluate how well the performance of the algorithm on the smaller class.

F-measure is a good indicator of the classifier’s performance since it takes into account both precision and recall.

1. Construct a trivial rule-based classifier that achieves better accuracy.

Rule based classifier - classify all as negative class:

{} -> -

Accuracy = 0.93

1. You are given the task of predicting whether there will be rain or sunshine tomorrow in Seattle. Results of your favorite classification algorithm on a test set of 1000 days are shown in the confusion matrix below.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
| Actual |  | Sunshine (+) | Rain (-) |
| Sunshine (+) | 80 | 70 |
| Rain (-) | 50 | 800 |

(i) Compute the precision, recall, F-measure, and accuracy for the confusion matrix.

     (Compute precision, recall, and the F-measure with respect to + class only.)

Accuracy = 880/1000 = 0.88

Precision = 80/(80+50) = 0.62

Recall = 80/(80+70) = 0.53

F-measure = (2 ∙ 80)/(2 ∙ 80+50+70) = 160/280 = 0.57

(ii) Which of these metrics is a poor indicator of the overall performance of your algorithm? Which of these metrics is the best indicator of the overall performance? Give a one sentence reason why this is the case?

Accuracy is a poor indicator of overall performance.

F-measure is a good indicator of overall performance.

There is a class imbalance problem (the sunshine (+) class is the minority), so accuracy is not a good indicator in this situation while the F-measure is an appropriate metric.

1. You are given the task of predicting whether the NASDAQ index will go up (+) or down (-) tomorrow. Results of your favorite classification algorithm on a test set of 1000 days are shown in the confusion matrix below.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | |
| Actual |  | + | - |
| + | 30 | 20 |
| - | 50 | 900 |

(i) Compute the accuracy, precision, recall, and F measure for the confusion matrix.

(Compute precision, recall, and the F-measure with respect to + class only.)

Accuracy = 930/1000 = 0.93

Precision = 30/(30+50) = 0.375

Recall = 30/(30+20) = 0.6

F-measure = (2 ∙ 30)/(2 ∙ 30+50+20) = 60/130 = 0.46

(ii) Which of these metrics is a poor indicator of the overall performance of your algorithm? Which of these metrics is a good indicator of the overall performance? Give a one sentence reason why this is the case?

Accuracy is a poor indicator of overall performance.

F-measure is a good indicator of overall performance.

There is a class imbalance problem (the + class is the minority), so accuracy is not a good indicator in this situation while the F-measure is an appropriate metric.

(iii) Construct a *trivial* rule-based model that gives better accuracy than your favorite algorithm above.

{ } → -

Accuracy = 950/1000 = 0.95

1. You are trying to evaluate two different blood tests, T1 and T2, that have been developed to detect a particular type of cancer.  T1 had been evaluated on a population of 200 subjects, out of which 100 were known to be suffering from cancer, while the remaining 100 were healthy. T2 had been evaluated on a different population of 1100 subjects, out of which 100 were known to be suffering from cancer, while the remaining 1000 were healthy. The results of these tests are shown in the following confusion matrices, along with the values of the following evaluation measures: TPR, FPR, Precision, the F-measure, and TPR/FPR.

**Test T1**

|  |  |  |
| --- | --- | --- |
| **Dataset:  (100 patients)** | **Predicted by Blood Test** | |
| **Actual** | **Cancer**  **(+ class)** | **No Cancer**  **(- class)** |
| **Cancer (+ class)** | 40 | 60 |
| **No Cancer (- class)** | 10 | 90 |

**Test T2:**

|  |  |  |
| --- | --- | --- |
| **Dataset:**  **(1100 patients)** | **Predicted by Blood Test** | |
| **Actual** | **Cancer**  **(+ class)** | **No Cancer**  **(- class)** |
| **Cancer (+ class)** | 40 | 60 |
| **No Cancer (- class)** | 55 | 945 |

1. Calculate the TPR, FPR, Precision, F1 score and TPR/FPR for the three tests.

Test 1: TPR: 0.4, FPR: 0.1, Precision: 0.8, F1-Score: 0.533, TPR/FPR: 4

Test 2: TPR: 0.4; FPR: 0.055, Precision: 0.42, F1-Score: 0.41, TPR/FPR: 7.27

1. According to the F-measure, which test is better?

Based on F-measure, T1 is better

1. According to TPR/FPR, which test is better?

Based on TPR/ FPR, T2 is better

d. For this situation, which evaluation measure (between F1-Score and TPR/FPR) should you use to make your selection between the two tests, T1 or T2? Why?

TPR/FPR. In this case, T2 is strictly better than T1 since both have the same TPR and T2 has lower FPR. Hence for any data set (irrespective of skew), T2 will outperform T1 on any reasonable measures."

Note that the value of F-measure depends on the skew of the data that is being used for evaluation. In particular, a given classifier will have different F-scores on two different datasets that have different ratio of positive and negative classes. Hence F-score should be used to compare two classifiers only if they are being tested on data sets with similar skew.

For this specific question, if classifiers T1 and T2 were both tested on the same data set (or on two data sets that have identical skew), thenT2 will have higher F-score than T1. The problem arises when we compare the F-score of T1 with F-score T2 but these scores are computed on data sets with different skew.

The value of TPR/FPR is independent of the skew in the data. Still, it can’t be used as a measure of choice in all situations. In fact, depending on the actual skew in the data, a classifier C1 with better TPR/FPR than C2 can have worse F-score relative to C2. However, if both classifiers have identical TPR (or FPR), then TPR/FPR can be used to select the best classifier (i.e., classifier with higher TPR/FPR will be superior according to any reasonable measure when the two classifiers are tested on the exact same data set).

e. Give an example scenario where you would reverse your choice of evaluation measure that you made in part (d). (That is, if you chose {TPR/FPR} in part (d), give an example of a scenario where you would prefer the F measure over TPR/FPR and vice-versa)

In a scenario where you know the skew in the target population matches that in the confusion matrix, the F measure can be preferred over {TPR/FPR}, as it captures both precision and recall for the target population, and thus allows you to identify a test that does not compromise one of these for the other.

1. You are trying to evaluate two different blood tests, T1 and T2, that have been developed to detect a particular type of cancer.  T1 had been evaluated on a population of 200 subjects, out of which 100 were known to be suffering from cancer, while the remaining 100 were healthy. T2 had been evaluated on a different population of 1000 subjects, out of which 100 were known to be suffering from cancer, while the remaining 900 were healthy. The results of these tests are shown in the following confusion matrices, along with the values of the following evaluation measures: TPR, FPR, Precision, the F-measure, and TPR/FPR.

**Test T1**

|  |  |  |
| --- | --- | --- |
| Dataset: (1000 patients) | **Predicted by Blood Test** | |
| **Actual** | **Cancer**  **(+ class)** | **No Cancer**  **(- class)** |
| **Cancer (+ class)** | 40 | 60 |
| **No Cancer (- class)** | 10 | 90 |

TPR: 0.4, FPR: 0.1, Precision: 0.8, F1-Score: 0.533, TPR/FPR: 4

**Test T2:**

|  |  |  |
| --- | --- | --- |
| Dataset: (1000 patients) | **Predicted by Blood Test** | |
| **Actual** | **Cancer**  **(+ class)** | **No Cancer**  **(- class)** |
| **Cancer (+ class)** | 40 | 60 |
| **No Cancer (- class)** | 50 | 850 |

TPR: 0.4; FPR: 0.055, Precision: 0.44, F1-Score: 0.41, TPR/FPR: 7.2

1. According to the F-measure, which test is better?

Based on the F-measure, T1 is better

1. According to TPR/FPR, which test is better?

Based on TPR/ FPR, T2 is better

c. For this situation, which evaluation measure (between F1-Score and TPR/FPR) should you use to make your selection between the two tests, T1 or T2? Why?

TPR/FPR. In this case, T2 is strictly better than T1 since both have the same TPR and T2 has lower FPR. Hence for any data set (irrespective of skew), T2 will outperform T1 on any reasonable measures."

Note that the value of F-measure depends on the skew of the data that is being used for evaluation. In particular, a given classifier will have different F-scores on two different datasets that have different ratio of positive and negative classes. Hence F-score should be used to compare two classifiers only if they are being tested on data sets with similar skew.

For this specific question, if classifiers T1 and T2 were both tested on the same data set (or on two data sets that have identical skew), thenT2 will have higher F-score than T1. The problem arises when we compare the F-score of T1 with F-score T2 but these scores are computed on data sets with different skew.

The value of TPR/FPR is independent of the skew in the data. Still, it can’t be used as a measure of choice in all situations. In fact, depending on the actual skew in the data, a classifier C1 with better TPR/FPR than C2 can have worse F-score relative to C2. However, if both classifiers have identical TPR (or FPR), then TPR/FPR can be used to select the best classifier (i.e., classifier with higher TPR/FPR will be superior according to any reasonable measure when the two classifiers are tested on the exact same data set).

d. Give an example scenario where you would reverse your choice of evaluation measure that you made in part (b)? (That is, if you chose {TPR/FPR} in part (c), give an example of a scenario where you would prefer {the F measure} over {TPR/FPR} and vice-versa)

Solution: In a scenario where you know the skew in the target population matches that in the confusion matrix, the F measure can be preferred over {TPR/FPR}, as it captures both precision and recall for the target population, and thus allows you to identify a test that does not compromise one of these for the other.

1. You are trying to evaluate two different intrusion detection systems, I1 and I2, that have been developed to detect a particular type of network intrusion.  I1 had been evaluated on an organization’s 2000 network log instances, out of which 1000 instances were known to be intrusions, while the remaining 1000 were safe instances. T2 had been evaluated on a different organization’s 100,000 network log instances, out of which 10,000 were known to be instances of intrusion, while the remaining 90,000 were safe instances. The results of these tests are shown in the following confusion matrices, along with the values of the following evaluation measures: TPR, FPR, Precision, the F-measure, and TPR/FPR.

**Evaluation on I1**

|  |  |  |
| --- | --- | --- |
| Dataset: (2000 instances) | **Predicted** | |
| **Actual** | **Intrusion**  **(+ class)** | **No Intrusion**  **(- class)** |
| **Intrusion (+ class)** | 400 | 600 |
| **No Intrusion (- class)** | 100 | 900 |

TPR: 0.4, FPR: 0.1, Precision: 0.8, F1-Score: 0.533, TPR/FPR: 4

**Evaluation on I2:**

|  |  |  |
| --- | --- | --- |
| Dataset: (100 000 instances) | **Predicted t** | |
| **Actual** | **Intrusion**  **(+ class)** | **No Intrusion**  **(- class)** |
| **Intrusion (+ class)** | 4000 | 6000 |
| **No Intrusion (- class)** | 5000 | 85000 |

TPR: 0.4; FPR: 0.055, Precision: 0.44, F1-Score: 0.41, TPR/FPR: 7.2

1. According to the F-measure, which Intrusion Detection System is better?

Based on the F-measure, I1 is better

1. According to TPR/FPR, which Intrusion Detection System is better?

Based on TPR/ FPR, I2 is better

c. For this situation, which evaluation measure (between F1-Score and TPR/FPR) should you use to make your selection between the two Intrusion Detection System, I1 and I2? Why?

TPR/FPR. In this case, T2 is strictly better than T1 since both have the same TPR and T2 has lower FPR. Hence for any data set (irrespective of skew), T2 will outperform T1 on any reasonable measures."

Note that the value of F-measure depends on the skew of the data that is being used for evaluation. In particular, a given classifier will have different F-scores on two different datasets that have different ratio of positive and negative classes. Hence F-score should be used to compare two classifiers only if they are being tested on data sets with similar skew.

For this specific question, if classifiers T1 and T2 were both tested on the same data set (or on two data sets that have identical skew), thenT2 will have higher F-score than T1. The problem arises when we compare the F-score of T1 with F-score T2 but these scores are computed on data sets with different skew.

The value of TPR/FPR is independent of the skew in the data. Still, it can’t be used as a measure of choice in all situations. In fact, depending on the actual skew in the data, a classifier C1 with better TPR/FPR than C2 can have worse F-score relative to C2. However, if both classifiers have identical TPR (or FPR), then TPR/FPR can be used to select the best classifier (i.e., classifier with higher TPR/FPR will be superior according to any reasonable measure when the two classifiers are tested on the exact same data set).

d. Give an example scenario where you would reverse your choice of evaluation measure that you made in part (b)? (That is, if you chose {TPR/FPR} in part (c), give an example of a scenario where you would prefer {the F measure} over {TPR/FPR} and vice-versa).

In a scenario where you know the skew in the target population matches that in the confusion matrix, the F measure can be preferred over {TPR/FPR}, as it captures both precision and recall for the target population, and thus allows you to identify a test that does not compromise one of these for the other.

1. You are trying to evaluate four different blood tests, T1, T2, T3, and T4, that have been developed to detect a particular type of cancer.  These tests have been developed by different organizations, and their evaluations by these organizations have been reported in the following confusion matrices, along with the values of the following evaluation measures: TPR, FPR, Precision, the F-measure, and TPR/FPR.

**Test T1:** TPR: 0.4, FPR: 0.1, Precision: 0.285, F1-Score: 0.33, TPR/FPR: 4

|  |  |  |
| --- | --- | --- |
| Dataset: (1100 patients) | **Predicted by Blood Test** | |
| **Actual** | **Cancer** **(+ class)** | **No Cancer**  **(- class)** |
| **Cancer (+ class)** | 40 | 60 |
| **No Cancer (- class)** | 100 | 900 |

**Test T2:** TPR: 0.2; FPR: 0.05, Precision: 0.4, F1-Score: 0.26, TPR/FPR: 4

|  |  |  |
| --- | --- | --- |
| Dataset: (200 patients) | **Predicted by Blood Test** | |
| **Actual** | **Cancer** **(+ class)** | **No Cancer**  **(- class)** |
| **Cancer (+ class)** | 20 | 80 |
| **No Cancer (- class)** | 5 | 95 |

**Test T3:** TPR: 0.5; FPR: 0.5, Precision: 0.5, F1-Score: 0.5, TPR/FPR: 1

|  |  |  |
| --- | --- | --- |
| Dataset: (200 patients) | **Predicted by Blood Test** | |
| **Actual** | **Cancer** **(+ class)** | **No Cancer**  **(- class)** |
| **Cancer (+ class)** | 50 | 50 |
| **No Cancer (- class)** | 50 | 50 |

**Test T4:** TPR: 0.5; FPR: 0.1, Precision: 0.833, F1-Score: 0.625, TPR/FPR: 5

|  |  |  |
| --- | --- | --- |
| Dataset: (200 patients) | **Predicted by Blood Test** | |
| **Actual** | **Cancer**  **(+ class)** | **No Cancer**  **(- class)** |
| **Cancer (+ class)** | 50 | 50 |
| **No Cancer (- class)** | 10 | 90 |

1. Between T1 and T3, which test is better? If you need more information to make a decision, what would it be?

T1 is better. T3 is random.

1. Between T1 and T2, which test is better? If you need more information to make a decision, what would it be?

Can’t say. Depends upon the distribution of positives and negatives in the population in which these tests will be applied

1. Between T1 and T4, which test is better? If you need more information to make a decision, what would it be?

T4 is better since it has the same FPR but better TPR

1. Between T4 and T3, which test is better? If you need more information to make a decision, what would it be?

T4 is better. T3 is random.

1. Between T4 and T2, which test is better? If you need more information to make a decision, what would it be?

Can’t say. The distribution of positives in the target population in which tests will be applied.

1. You are working with a doctor to evaluate how well a new, inexpensive blood test can detect a particular type of cancer. To assess its effectiveness, the blood test was conducted on a population of 1000 subjects, out of which 100 are known to be suffering from cancer while the remaining 900 are healthy. The results of the test are shown in the following confusion matrix:

|  |  |  |
| --- | --- | --- |
| Dataset: (1000 patients) | **Predicted by Blood Test** | |
| **Actual** | **Cancer  (+ class)** | **No Cancer  (- class)** |
| **Cancer (+ class)** | 90 | 10 |
| **No Cancer (- class)** | 90 | 810 |

a. Considering cancer to be the positive class, compute precision, TPR (recall), and FPR for the above confusion matrix.

Precision = 90/(90 + 90) = 0.50, TPR = 90/(10 + 90) = 0.90

FPR = 90/(90 + 810) = 0.1

b. The doctor wants to conduct the same experiment on a sample of the population from a different city that could potentially have a different proportion of cancer patients and healthy subjects. In order to compare the results of the two studies, which of the following metric pairs would you prefer: {TPR,FPR} or {precision, recall}? Provide a brief justification.

Solution: {TPR,FPR}as they are invariant to the class imbalance in the data. As a result, TPRs and FPRs obtained from the two studies would still be comparable.

c. Give an example of a class-imbalance classification problem where you would reverse your choice of metric pair that you made in part (b)? (That is, if you chose {TPR, FPR} in part (b), give an example of a scenario where you would prefer {precision, recall} over {TPR,FPR} and vice-versa).

Solution: Applications where {precision, recall} could be preferred over {TPR, FPR} are those where test data contains all of the population of interest (or if the test sample contains positive and negative samples in the same proportion as the entire population) - some examples are information retrieval, fraud detection. If the testing is done on a random sample of positives and another random sample of negatives (which is quite often the case in many applications, including health care), then precision can change dramatically depending upon the skew between positive and negative samples chosen for testing.

Note:

Alternative solution: Based on the assumption that test data contains all of the population of interest, the choice can be changed according to different application purposes. If predicting positive class is much more significant than negative class, or if predicting negative class is not interesting, {precision, recall} can be preferred over {TPR, FPR}. On the contrary, if positive and negative classes are equally important, then we may prefer {TPR, FPR}.

Incorrect / partially correct answers:

a) "when the amount of the two classes are balanced." The question has emphasized "class-imbalance classification problem."

b) "If the class proportions keep unchanged between training and test data (i.e., reverse the condition in Q1b)." The test data is required to contain the same proportion as the entire population, not training data. And {TPR, FPR} might still be preferred if both the classes are equally interesting.

c) "{TPR, FPR} is preferred for comparing randomly guessing classifiers." The measure choice should depend on the application, rather than properties of classifiers. But you can say, "if classifiers are randomly guessing, {TPR,FPR} can capture it and help us to discard them."

1. Consider a test data consisting of 1000 samples with two classes: +ve class (100 samples) and –ve class (900 samples). We want to compare two classifiers: C1 and C2. Classifier C1 classifies test data to +ve class randomly with a probability 0.5 and classifier C2 classifies test data to +ve class randomly with a probability 0.8. The confusion matrices for C1 and C2 are provided below.

|  |  |  |
| --- | --- | --- |
| Classifier C1 | Predicted ‘+’ | Predicted ‘-’ |
| Actual ‘+’ | 50 | 50 |
| Actual ‘-’ | 450 | 450 |

|  |  |  |
| --- | --- | --- |
| Classifier C2 | Predicted ‘+’ | Predicted ‘-’ |
| Actual ‘+’ | 80 | 20 |
| Actual ‘-’ | 720 | 180 |

The performance of the two classifiers has been summarized in the following table.

|  |  |  |
| --- | --- | --- |
|  | C1 | C2 |
| Precision | 0.1 | 0.1 |
| Recall | 0.5 | 0.8 |
| TPR | 0.5 | 0.8 |
| FPR | 0.5 | 0.8 |

Based on your observations from these results, would you consider C2 to be a better classifier than C1? Which of the following two evaluation metric pairs: {precision, recall} and {TPR, FPR}, did you choose to arrive at your result? Provide a brief justification.

Answer: No C2 is not a better classifier than C1 since they both are performing random guessing with different probabilities of assigning an instance to the positive class at random. I used {TPR, FPR} to arrive at this result. Both C1 and C2 have TPR = FPR, which corresponds to the random guessing baseline in the ROC curve.

1. A classifier is being tested on two datasets: Dataset 1, with 100 positives and 100 negatives, and Dataset 2, with 100 positives and 500 negatives. The confusion matrices of the classifier on the two datasets are provided below.

|  |  |  |
| --- | --- | --- |
| Dataset 1 | Predicted ‘+’ | Predicted ‘-’ |
| Actual ‘+’ | 80 | 20 |
| Actual ‘-’ | 20 | 80 |

|  |  |  |
| --- | --- | --- |
| Dataset 2 | Predicted ‘+’ | Predicted ‘-’ |
| Actual ‘+’ | 80 | 20 |
| Actual ‘-’ | 100 | 400 |

The performance of the classifier on the two datasets has been summarized in the following table.

|  |  |  |
| --- | --- | --- |
|  | Dataset 1 | Dataset 2 |
| Precision | 0.8 | 0.44 |
| Recall | 0.8 | 0.8 |
| TPR | 0.8 | 0.8 |
| FPR | 0.2 | 0.2 |

Based on your observations from these results, if you had to choose between the following two evaluation metric pairs: {precision, recall} and {TPR, FPR}, which one would you choose in the following scenarios and why? Provide brief explanations in context with your observations from the results above.

1. The evaluation is required to be invariant to changes in the relative numbers of positives and negatives in the evaluation dataset.

Answer: {TPR, FPR}, since the TPR and FPR values can be seen to be the same in Dataset 1 and Dataset 2, which have different relative numbers of positives and negatives.

1. The number of false positive predictions must be minimized.

Answer: {Precision, Recall}, since precision measures the number of false positives (falsely detected as positive by the classifier, or in other words, false predictions of the positive class) relative to the total number of predictions. FPR, on the other hand, measures the number of false positives relative to the total number of negatives, which can still be small in class imbalance scenarios even when the absolute number of false positives is large relative to the total number of positive predictions, as observed in Dataset 2.

1. Compute the accuracy of the classifier on Dataset 2 (you can leave your answer in fractions). Construct a trivial classifier that can achieve better accuracy on Dataset 2 without even looking at the attributes of the data. What is the accuracy of this trivial classifier?

Answer: Accuracy of classifier on Dataset 2 = 480/600 = 4/5. A trivial classifier that achieves better accuracy on Dataset 2 is one that assigns every instance to the negative class, regardless of its attributes. Accuracy of this trivial classifier on Dataset 2 = 500/600 = 5/6.

1. A classifier is being tested on two datasets: (a) Dataset 1, with 100 positives and 100 negatives, and (b) Dataset 2, with 100 positives and 900 negatives. The confusion matrices of the classifier on the two datasets are provided below.

|  |  |  |
| --- | --- | --- |
| Confusion Matrix for **Dataset 1** | Predicted Label = Positive | Predicted Label = Negative |
| True Label = Positive | 90 | 10 |
| True Label = Negative | 10 | 90 |
| Confusion Matrix for **Dataset 2** | Predicted Label = Positive | Predicted Label = Negative |
| True Label = Positive | 90 | 10 |
| True Label = Negative | 90 | 810 |

The performance of the classifier on the two datasets has been summarized in the following table.

|  |  |  |
| --- | --- | --- |
|  | Dataset 1 | Dataset 2 |
| Precision | 0.9 | 0.5 |
| Recall | 0.9 | 0.9 |
| TPR | 0.9 | 0.9 |
| FPR | 0.1 | 0.1 |

Based on your observations from these results, if you had to choose between the following two evaluation metric pairs: {precision, recall} and {TPR, FPR}, which one would you choose in the following scenarios and why? Provide brief explanations in context with your observations from the results above.

1. The evaluation is required to be invariant to changes in the relative numbers of positives and negatives in the evaluation dataset.

{TPR, FPR}, since the TPR and FPR values can be seen to be the same in Dataset 1 and Dataset 2, which have different relative numbers of positives and negatives.

1. The number of false positive predictions has to be minimized.

{Precision, Recall}, since precision measures the number of false positives (falsely detected as positive by the classifier, or in other words, false predictions of the positive class) relative to the total number of predictions. FPR, on the other hand, measures the number of false positives relative to the total number of negatives, which can still be small in class imbalance scenarios even when the absolute number of false positives is large relative to the total number of positive predictions, as observed in Dataset 2.

1. You are working with a doctor to evaluate how well a new, inexpensive blood test can detect a particular type of cancer. 1000 subjects are recruited from a population at high risk for cancer and evaluated for cancer using a very expensive, but 100% accurate medical procedure. 100 subjects are found to have cancer. The 100 subjects with cancer and another 100 subjects without cancer are given the inexpensive blood test. Results are shown in the following confusion matrix.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted by Blood Test | |  |
| Actual | Cancer | No Cancer |  |
| Cancer | 90 | 10 | 100 |
| No Cancer | 10 | 90 | 100 |
|  | 100 | 100 | 200 |

Cancer: precision = 90/(90 + 10) = 0.90, recall = TPR = 90/(10 + 90) = 0.90

No Cancer: precision = 90/(10 + 90) = 0.90, recall = 90/(10 + 90) = 0.9, FPR = 10/(10 + 90) = 0.10

The doctor is very excited about these results but wants to see what the results will be after all the blood tests are evaluated. The remaining 800 subjects (none of which have cancer) are given the blood test.

The confusion matrix for all 1000 subjects is given below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted by Blood Test | |  |
| Actual | Cancer | No Cancer |  |
| Cancer | 90 | 10 | 100 |
| No Cancer | 90 | 810 | 900 |
|  |  |  |
|  | 180 | 820 | 1000 |

Cancer: precision = 90/(90 + 90) = 0.50, recall = TPR = 90/(10 + 90) = 0.90

No cancer: precision = 810/(10 + 810) = 0.988, recall = 810/(90 + 810) = 0.9, FPR = 90/(90 + 810) = 0.1

a. Which of the measures (precision, recall, TPR, FPR) have changed and which have stayed the same? Comment on why some measures were affected and others were not.

Solution: Recall for both classes, as well as TPR and FPR, remained the same. Precision for both classes changed. Even though the classifier was still performing the same as measured by TPR and FPR, precision is sensitive to the relative number of members of each class

b. You were disappointed by the change you observed in one of the measures from the first confusion matrix to the second, but the doctor was not. The doctor tells you to compute sensitivity (recall of the positive class) and specificity (1-FPR) for both confusion matrices. Note that cancer is the positive class. Comment on why these measures are preferred in the medical profession.

Solution: Sensitivity = 0.90 for both tables, Specificity = 0.90 for both tables. These measures are preferred because, unlike precision, they are not sensitive to the relative number of subjects in each class. Thus, if a drug is evaluated at two different sites, but with different relative numbers in each class, the results will still be comparable

c. When might precision be the preferred measure of the classification performance?

Solution: Precision is useful when the ratio of (+) and (-) in the test sample is similar to the actual population in which the test is to be deployed. Another situation where precision is essential is when the number of false alarms must be minimized, such as in fraud detection.

1. You are given a task to evaluate how well a new fire mapping algorithm works. The fire mapping algorithm is a Bayesian classifier that labels all the locations into two classes, burned (+) and unburned (-). To evaluate the algorithm, two regions are tested. The confusion matrices of these two regions are given in the tables below.

|  |  |  |  |
| --- | --- | --- | --- |
| Data set 1 | | Predicted class | |
| Burned | Unburned |
| Actual  Class | Burned | 30 | 20 |
| Unburned | 10 | 40 |

|  |  |  |  |
| --- | --- | --- | --- |
| Data set 2 | | Predicted class | |
| Burned | Unburned |
| Actual  Class | Burned | 30 | 20 |
| Unburned | 1000 | 4000 |

a) Calculate the TPR, FPR, Precision and Recall for the burned (+) class for both these data sets.

TPR(DS1) = 30/50 = 0.6 TPR(DS2) = 30/50 = 0.6

FPR(DS1) = 10/(10+40) = 0.2 FPR(DS2) = 1000/(1000+4000) = 0.2

Precision(DS1) = 30/(30+10) = 3/4 Precision(DS2) = 30/(30+1000) = 3/103

Recall(DS1) = 30/(30+20) = 0.6 Recall(DS2) = 30/(30+20) = 0.6

b) Is there a difference in their values for the two data sets? If so, what characteristics of the data sets (that are used to derive the above contingency tables) lead to the differences between the values of (TPR,FPR) and (Precision, Recall) that you observe above.

FPR and TPR (and also Recall) are the same for the two datasets. However, Precision is different. The class skew between the two classes is significantly different. Since (TPR, FPR) are invariant to class skew, they don’t change. However, Precision is not invariant to class skew, and thus its value changes.

c) Compute Accuracy and F-measure with respect to ‘burned’ class for dataset 2.

Accuracy (DS2) = 4030/5050 = 0.798

F-measure (DS2) = 2\*3/103\*0.6 /(3/103+0.6) = 0.056

d) Comment on the performance of the given algorithm. Your answer should include 1) which evaluation metric (F-measurement or accuracy) is chosen. 2) Why you choose it. 3) What is your conclusion (Does the new algorithm work well or not) based on the evaluation metric?

I choose F-measure. In the case of imbalanced class problem, accuracy does not do a good job of evaluating the performance of the algorithm on the smaller class. The F-measure is small. Thus, the algorithm does not perform well.

e) Is it possible to construct a trivial rule-based classifier with better i) F-measure? ii) accuracy? If yes, construct the classifier.

1. No.
2. Yes. Label all the locations as unburned. Then, accuracy is 5000/5050 > accuracy (DS2).
3. You are given a Bayesian classification algorithm that is able to identify if a forest location has been burned or not using its satellite observations. The algorithm labels all locations into two classes, burned and unburned. Further, it is known that the number of unburned locations is much larger than the number of burned locations since forest fires are rare events. The algorithm was evaluated on a dataset and the obtained confusion matrix is shown in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| Results of the Bayesian fire classification algorithm | | Predicted class | |
| Burned | Unburned |
| Actual Class | Burned | 30 | 20 |
| Unburned | 1000 | 4000 |

The performance of the Bayesian algorithm on this dataset can be summarized with the help of the following evaluation measures, where the positive class has been chosen to be the burned class:

|  |  |
| --- | --- |
| TPR / Recall | 0.6 |
| FPR | 0.2 |
| Precision | 0.029 |
| Accuracy | 0.798 |
| F-Measure | 0.056 |

Using the above information, answer the following questions:

a. Comment on the performance of the algorithm on the given dataset. Your answer should include 1) which evaluation metric (F-measurement or accuracy) is chosen. 2) Why you choose it. 3) What is your conclusion (does the new algorithm work well or not) based on the evaluation metric?

Choose F-measure. In the case of an imbalanced class problem, accuracy does not give a good evaluation of the performance of the algorithm on the smaller class. The F-measure on the *burned* class is low. Thus, the algorithm does not perform well.  
  
b. Can you construct a trivial rule-based classifier that achieves better accuracy than the given algorithm? If yes, what is the classifier and what is the resulting accuracy obtained by it?

Label all the locations as unburned. Then, accuracy is 5000/5050 > 0.798.

c. Can you construct a trivial rule-based classifier that achieves better F-measure than the given algorithm? If yes, what is the classifier and what is the resulting F-measure obtained by it?

No such classifier exists.

1. Consider the confusion matrices of a classifier, M, for Dataset 1 and Dataset 2:

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset 1 | | Predicted class | |
|  | | + | - |
| Actual class | + | 45 | 5 |
| - | 10 | 40 |
|  |
| Dataset 2 | | Predicted class | |
|  | | + | - |
| Actual class | + | 45 | 5 |
| - | 100 | 400 |
|  |

a)  Calculate the TPR and FPR of M for the + class for both these datasets. Is there a difference in their values for the two datasets?

Solution: TPR(DS1) = 45/(45+5) = 0.9, FPR(DS1) = 10/(10+40)=0.2 TPR(DS2) = 45/(45+5) = 0.9, FPR(DS2) = 100/(100+400)=0.2

TPR and FPR are the same for M for the two data sets.

b)  Now, calculate Precision and Recall of M for the + class for both these datasets. Is there a difference in their values for the two datasets?

Solution: P(DS1) = 45/(45+10)=0.818, R(DS1)=45/(45+5)=0.9 P(DS2)=45/(45+100)=0.31, R(DS2)=45/(45+5)=0.9

Although the recall is the same for the two data sets, precision is significantly different.

c) What characteristics of the datasets (that are used to derive the above contingency tables) lead to the differences (if any) between the values of (TPR, FPR) and (Precision, Recall) that you observe above.

Solution: The class imbalance between the + and – classes is significantly different between the two data sets, i.e., 1:1 for dataset 1 and 1:10 for dataset 2. Since (TPR, FPR) are invariant to class imbalance (skewness), they don’t change. However, Precision is not invariant to class imbalance, and hence its value changes.

1. Consider a data set with instances belonging to one of two classes - positive (+) and negative (-). A classifier was built using a training set consisting of an equal number of positive and negative instances. Among the training instances, the classifier has a recall of m=50% on the positive class and a recall of n=90% on the negative class.

The trained classifier is now tested on two data sets. Both have similar data characteristics as the training set. The first data set has 1000 positive and 1000 negative instances. The second data set has 100 positive and 1000 negative instances.

A. Draw the expected confusion matrix summarizing the *expected* classifier performance on the two data sets.

|  |  |  |
| --- | --- | --- |
|  | Algorithm Output = (+) | Algorithm Output = (-) |
| True Label = (+) | 1000\*0.5 = 500 | 1000\*0.5 = 500 |
| True label = (-) | 1000\*0.1 = 100 | 1000\*0.9 = 900 |

|  |  |  |
| --- | --- | --- |
|  | Algorithm Output = (+) | Algorithm Output = (-) |
| True Label = (+) | 100\*0.5 = 50 | 100\*0.5 = 50 |
| True label = (-) | 1000\*0.1 = 100 | 1000\*0.9 = 900 |

B. What is the accuracy of the classifier on the training set? Compute the precision, TPR and FPR for the two test data sets using the confusion matrix from part A. Also report the accuracy of the classifier on both data sets.  
  
Training accuracy = (0.5+\*0.9)/2 = 0.7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | TPR | FPR | Accuracy |
| Data set 1 | 500/600 = 0.83 | 0.5 | 0.1 | (500+900)/2000 = 0.7 |
| Data set 2 | 50/150 = 0.33 | 0.5 | 0.1 | (50+900)/1100 = 0.86 |

C. In the scenario where the class imbalance is pretty high, how are precision and recall better metrics in comparison to overall accuracy? What information does precision capture that recall doesn’t?  
  
When the skew gets pretty high - i.e. the total number of negatives are much more than the number of positives, the overall accuracy of the classifier is mainly dominated by the performance of the classifier on the majority negative class. From the curve in part C, we can see that the accuracy is almost constant after the skew increases beyond a certain point. If a classifier does well on the majority negative class but does poorly on the positive class, the overall accuracy wouldn’t be able to capture this.

Precision is important to consider because it keeps track of the number of false positives of the algorithm and how significant are they in comparison to the true positives.  If your algorithm flags 1000 instances as positive, but only 100 of them are truly positives, its performance is not good even if it captures all the truly positive instances, thereby giving it a high recall.

1. Consider a data set with instances belonging to one of two classes – positive (+) and negative (-). A classifier was built using a training set consisting of 100 positive and 100 negative instances. Among the training instances, the classifier correctly classifies 90 % of the positive class instances as positive and incorrectly classifies 10 % of the negative class instances as positive. The trained classifier is tested on two data sets. Both have similar data characteristics as the training set, in the sense that the classification performance (in terms of TPR and FPR) is maintained when applied to the two datasets. The first data set has 1000 positive and 1000 negative instances. The second data set has 100 positive and 1000 negative instances.

A. Fill up the expected confusion matrix summarizing the ***expected*** classifier performance for both the datasets.

**Dataset 1. 1000 positive, 1000 negative**

|  |  |  |
| --- | --- | --- |
| Dh | Algorithm Output = (+) | Algorithm Output = (-) |
| True Label = (+) | 900 | 100 |
| True label = (-) | 100 | 900 |

**Dataset 2. 100 positive, 1000 negative**

|  |  |  |
| --- | --- | --- |
|  | Algorithm Output = (+) | Algorithm Output = (-) |
| True Label = (+) | 90 | 10 |
| True label = (-) | 100 | 900 |

B. Which of these two data sets have better values for recall, precision, and F measure? Answer this by filling in the boxes given below with ‘>’, ‘<’ or ‘=’. No explanation required.

**Hint:** You don’t need to compute the values explicitly to answer this question

1. Dataset 1: Recall of positive class \_\_\_=\_\_\_\_ Dataset 2: Recall of positive class
2. Dataset 1: Precision of positive class \_\_\_>\_\_\_\_ Dataset 2: Precision of positive class
3. Dataset 1: F-measure \_\_\_>\_\_\_\_ Dataset 2: F-measure
4. Consider a data set with instances belonging to one of two classes: positive (+) and negative (-).   
   Among the training instances, the classifier correctly classifies 2/3 of the positive class instances as positive and incorrectly classifies 1/3 of the negative class instances as positive. The trained classifier is tested on two data sets. Both have similar data characteristics as the training set, in the sense that the classification performance (in terms of TPR and FPR) is maintained when applied to the two datasets. The first data set has 300 positive and 300 negative instances. The second data set has 75 positive and 300 negative instances.

A. Fill up the expected confusion matrix summarizing the expected classifier performance for both the datasets.

**Dataset 1. 300 positive, 300 negative**

|  |  |  |
| --- | --- | --- |
| Dh | Algorithm Output = (+) | Algorithm Output = (-) |
| True Label = (+) | 200 | 100 |
| True label = (-) | 100 | 200 |

**Dataset 2. 75 positive, 300 negative**

|  |  |  |
| --- | --- | --- |
|  | Algorithm Output = (+) | Algorithm Output = (-) |
| True Label = (+) | 50 | 25 |
| True label = (-) | 100 | 200 |

**B.** Calculate recall, precision, and F measure for the two datasets above. Leave answers as a fraction if simplifying is hard without a calculator.

1. Dataset 1: Recall of positive class \_\_\_\_ Dataset 2: Recall of positive class \_\_\_\_
2. Dataset 1: Precision of positive class \_\_\_ Dataset 2: Precision of positive class \_\_\_\_
3. Dataset 1: F-measure \_\_\_ Dataset 2: F-measure \_\_\_\_

Dataset1: Recall: 2/3, Precision: 2/3, F-measure: 2/3

Dataset2: Recall: 2/3, Precision: 1/3, F-measure: 2\*2/3\*1/3 / (2/3+1/3) = 4/9

**C.** Which of the 3 measures (recall, precision, and F-measure) changed from one data set to another? Why?

Precision and the F-measure because the ratio of positives and negatives changed.

1. You are asked to evaluate the performance of two classification models, M1 and M2, for a binary classification problem with classes ‘+’ and ‘–’. For every test instance, x, each of the two models provides a posterior probability of x belonging to class ‘+’. The table below provides a list of 10 test instances with their true classes, and their posterior probabilities of belonging to class ‘+’, according to M1 and M2. Assume that we are mostly interested in detecting instances from the positive class.

|  |  |  |  |
| --- | --- | --- | --- |
| Instance | True Class | P(+|M1) | P(+|M2) |
| 1 | + | 0.98 | 0.27 |
| 2 | + | 0.31 | 0.45 |
| 3 | + | 0.92 | 0.95 |
| 4 | + | 0.31 | 0.46 |
| 5 | + | 0.93 | 0.23 |
| 6 | - | 0.33 | 0.13 |
| 7 | - | 0.47 | 0.08 |
| 8 | - | 0.46 | 0.19 |
| 9 | - | 0.24 | 0.37 |
| 10 | - | 0.45 | 0.04 |

Table

1. Plot the ROC curve for both M1 and M2. (You should plot them on the same graph.) Which model do you think is better? Explain your reasons

Solution: ROC Curve for M2 (red) and M1(blue) is shown below

A screenshot of a cell phone

Description automatically generatedrain

The area under the curve (AUC) of M1 = 0.68 and the AUC of M2 = 0.92. Since M2 has a higher AUC than M1, M2 is a better model. Also, notice that for any given FPR, M2 has better (or same) TPR than M.

b) Suppose you choose a cutoff threshold to be t = 0.4 for both the models, M1 and M2. In other words, any test instance whose posterior probability is greater than t will be classified as a positive example. Compute the Precision, Recall, and F-Measure for M1 and M2 after using the cutoff threshold of t. Which model is better using F-measure as the evaluation criterion? Are the results consistent with what you expect from the ROC curve?

Solution: For M1 at t = 0.4 threshold, Precision = 0.5, Recall = 0.6, F-Measure = 0.54.

For M2 at t = 0.4 threshold, Precision = 1, Recall = 0.6, F-Measure = 0.75.

By looking at the F-Measure values, we conclude that M2 is a better model since it has a higher F-Measure than M1. This result is consistent with the result in (a), by using AUC of the ROC curve as the evaluation criterion.

c)  Repeat part (b) using t = 0.7. Which model is better using F-measure as the evaluation criterion? Are the results consistent with what you expect from the ROC curve?

Solution: For M1 at t = 0.7 threshold, Precision = 1, Recall = 0.6, F-Measure = 0.75.

For M2 at t = 0.7 threshold, Precision = 1, Recall = 0.2, F-Measure = 0.33.

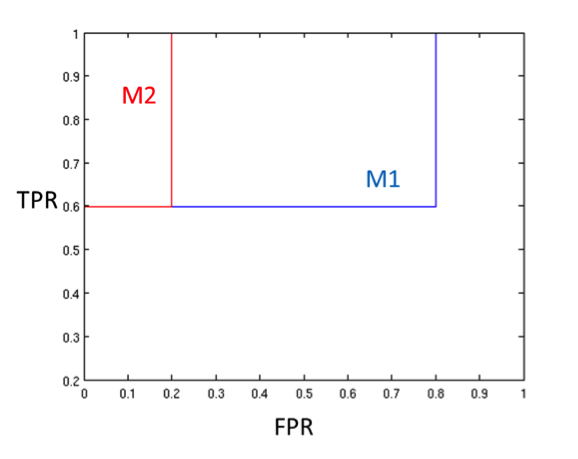
By looking at the F-Measure values, we conclude that M1 is a better model since it has a higher F-Measure than M2. This result is inconsistent with the result in (a), by using the AUC of the ROC curve as the evaluation criterion. This indicates that evaluating a classifier by computing its F-Measure at a given threshold can give a different result than by constructing its ROC curve and measuring its AUC.

1. You are asked to evaluate the performance of two classification models, M1 and M2, for a binary classification problem with classes ‘+’ and ‘–’. For every test instance, x, each of the two models provides a posterior probability of x belonging to class ‘+’. The table below provides a list of 10 test instances with their true classes, and their posterior probabilities of belonging to class ‘+’, according to M1 and M2. Assume that we are mostly interested in detecting instances from the positive class.

|  |  |  |  |
| --- | --- | --- | --- |
| Instance | True Class | P(+|M1) | P(+|M2) |
| 1 | + | 0.94 | 0.27 |
| 2 | + | 0.31 | 0.45 |
| 3 | + | 0.76 | 0.95 |
| 4 | + | 0.31 | 0.46 |
| 5 | + | 0.82 | 0.23 |
| 6 | - | 0.33 | 0.13 |
| 7 | - | 0.47 | 0.08 |
| 8 | - | 0.46 | 0.19 |
| 9 | - | 0.24 | 0.37 |
| 10 | - | 0.45 | 0.04 |

Table 1

1. Plot the ROC curve for both M1 and M2. (You should plot them on the same graph.) Which model do you think is better? Explain your reasons.  
     
   Solution: ROC Curve for M2 (red) and M1(blue) is shown below

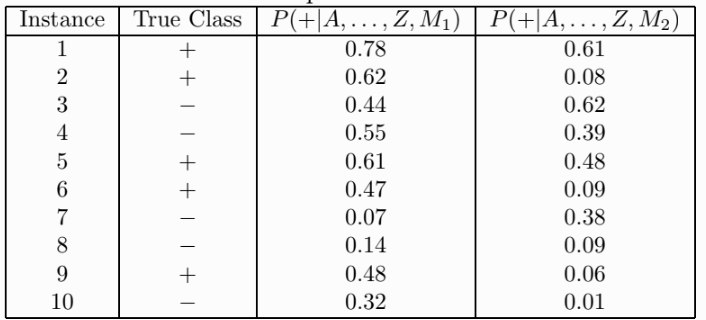


The area under the curve (AUC) of M1 = 0.68 and the AUC of M2 = 0.92. Since M2 has a higher AUC than M1, M2 is a better model. Also, notice that for any given FPR, M2 has better (or same) TPR than M1.

b) Suppose you choose a cutoff threshold to be t = 0.4 for both the models, M1 and M2. In other words, any test instance whose posterior probability is greater than t will be classified as a positive example. Compute the Precision, Recall, and F-Measure for M1 and M2 after using the cutoff threshold of t. Which model is better using F-measure as the evaluation criterion? Are the results consistent with what you expect from the ROC curve?   
Solution: For M1 at t = 0.4 threshold, Precision = 0.5, Recall = 0.6, F-Measure = 0.54. For M2 at t = 0.4 threshold, Precision = 1, Recall = 0.6, F-Measure = 0.75. By looking at the F-Measure values, we conclude that M2 is a better model, since it has a higher F-Measure than M1. This result is consistent with the result in (a), by using AUC of the ROC curve as the evaluation criterion.

c) Repeat part (b) using t = 0.7. Which model is better using F-measure as the evaluation criterion? Are the results consistent with what you expect from the ROC curve?   
  
Solution: For M1 at t = 0.7 threshold, Precision = 1, Recall = 0.6, F-Measure = 0.75. For M2 at t = 0.7 threshold, Precision = 1, Recall = 0.2, F-Measure = 0.33. By looking at the F-Measure values, we conclude that M1 is a better model since it has a higher F-Measure than M2. This result is inconsistent with the result in (a), by using the AUC of the ROC curve as the evaluation criterion. This indicates that evaluating a classifier by computing its F-Measure at a given threshold can give a different result than by constructing its ROC curve and measuring its AUC.

1. You are asked to evaluate the performance of two classification models, M1 and M2. The test set you have chosen contains 26 binary attributes, labeled as A through Z. The table below shows the posterior probabilities obtained by applying the models to the test set. (Only the posterior probabilities for the positive class are shown). As this is a two-class problem, P(−) = 1 − P(+) and P(−|A, . . ., Z) = 1 − P(+|A, . . ., Z). Assume that we are only interested in detecting instances from the positive class.

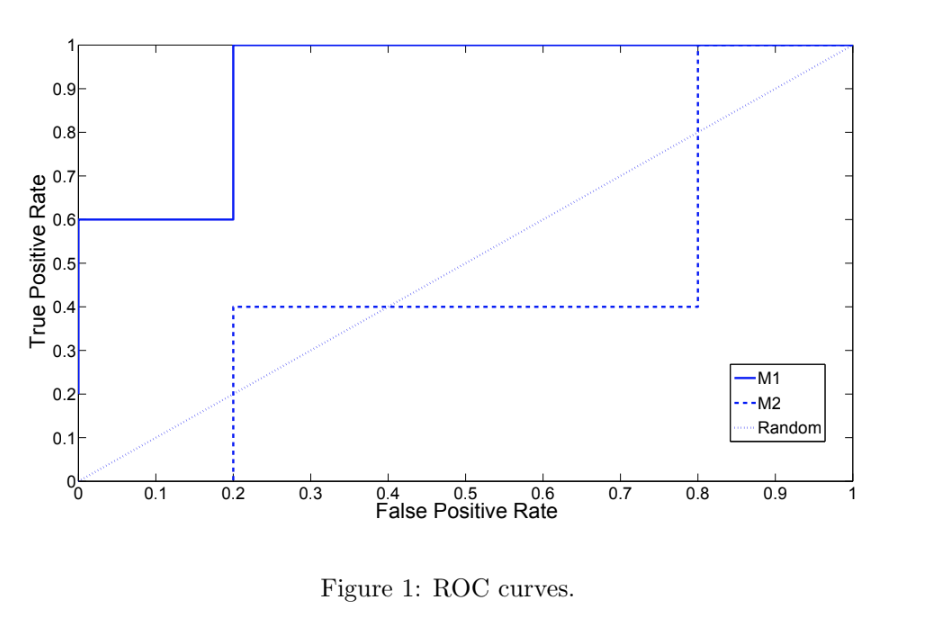


**(a)** Plot the ROC curve for both M1 and M2. (You should plot them on the same graph.

Clearly, hand-drawn plots are acceptable.) Which model do you think is better? Explain

your reasons.

The ROC curves for M1 and M2 are shown in the Figure below.



M1 is better since its area under the ROC curve is larger than the area under ROC

curve for M2.

**(b)** For model M1, suppose you choose the cutoff threshold to be t = 0.5. In other words, any

test instances whose posterior probability is greater than t will be classified as a positive

example. Compute the precision, recall, and F-measure obtained from the model at this

threshold value.

When t = 0.5, the confusion matrix for M1 is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **+** | **-** |
| Actual | **+** | 3 | 2 |
|  | **-** | 1 | 4 |

Precision = 3/4 = 75%.

Recall = 3/5 = 60%.

F-measure = (2 × 0.75 × .6)/(0.75 + .6) = 0.667.

**(c)** Repeat the analysis for part (b) using the same cutoff threshold on model M2. Compare

the F-measure results obtained from both the models. Which model is better? Are the

results consistent with what you expect from the ROC curve?

When t = 0.5, the confusion matrix for M2 is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **+** | **-** |
| Actual | **+** | 1 | 4 |
|  | **-** | 1 | 4 |

Precision = 1/2 = 50%.

Recall = 1/5 = 20%.

F-measure = (2 × .5 × .2)/(0.5 + .2) = 0.2857.

Based on F-measure, M1 is still better than M2. This result is consistent with the ROC

Plot.

**(d)** Repeat part (b) for model M1 using the threshold t = 0.1. Based on the F-measure,

which threshold do you prefer, t = 0.5 or t = 0.1?

When t = 0.1, the confusion matrix for M1 is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **+** | **-** |
| Actual | **+** | 5 | 0 |
|  | **-** | 4 | 1 |

Precision = 5/9 = 55.55%.

Recall = 5/5 = 100%.

F-measure = (2 × .5555 × 1)/(.5555 + 1) = 0.715.

According to F-measure, t = 0.1 is better than t = 0.5.

**(e)** Which threshold, t = 0.5 or t = 0.1, would you use as per the results of the ROC

analysis?

Using the confusion matrices above, it can be seen that when t = 0.1, FPR = 0.8 and TPR = 1 On the other hand, when t = 0.5, FPR = 0.2 and TPR = 0.6. Since

(0.2, 0.6) is closer to the point (0, 1), we favor t = 0.5.

**(f)** Are the choices of thresholds consistent based on the F-measure and ROC analyses?

Briefly comment on why or why not?

The choices of thresholds based on the two analyses are inconsistent, primarily since F-measure and ROC are different ways of evaluating the performance of a classifier.

We can also show this by computing the area under the ROC curve

For t = 0.5, area = 0.6 × (1 − 0.2) = 0.6 × 0.8 = 0.48.

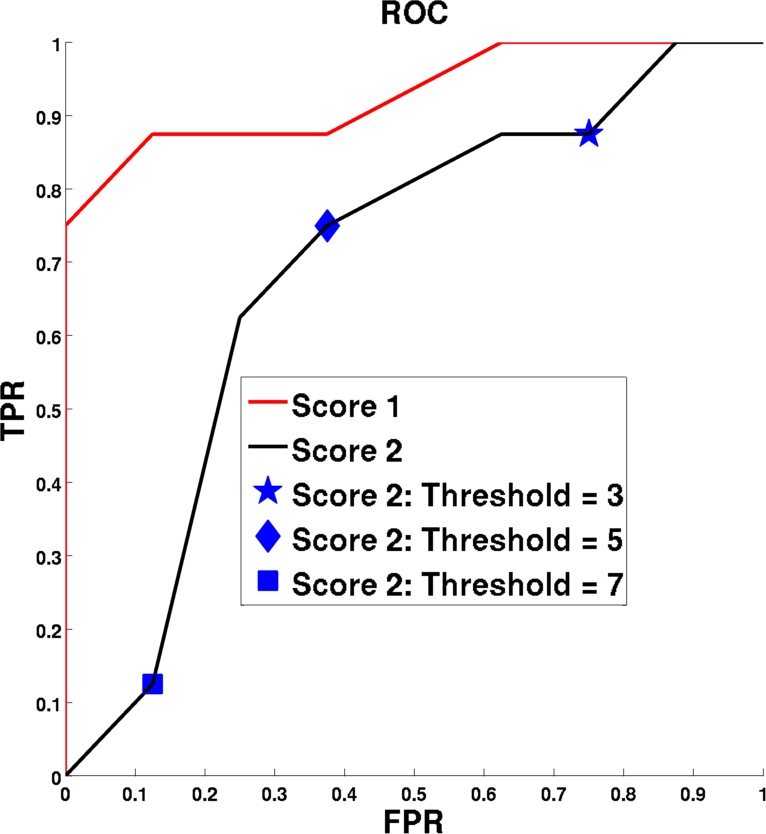
For t = 0.1, area = 1 × (1 − 0.8) = 1 × 0.2 = 0.2.

Since the area for t = 0.5 is larger than the area for t = 0.1, we prefer t = 0.5.

1. Consider two classification algorithms that output real-valued scores. The class label prediction (positive or negative) is obtained by thresholding this score. The instances input to the algorithm and the corresponding scores are given below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instance ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| True Class Label | + | + | - | - | - | + | - | + | + | + | + | - | - | + | - | - |
| Score 1 | 7.9 | 8.2 | 1.2 | 2.8 | 2.5 | 9.2 | 5.8 | 6.3 | 8.9 | 5.2 | 2.1 | 3.2 | 4.5 | 6.2 | 1.9 | 0.7 |
| Score 2 | 7.6 | 6.2 | 1.7 | 2.4 | 3.5 | 4.2 | 5.9 | 6.3 | 6.9 | 5.7 | 2.8 | 4.3 | 6.5 | 6.1 | 7.5 | 4.3 |

1. Draw the ROC curve for the given data for both algorithms, thresholding the scores from 0 to 10, taking steps of 1. Note: For a threshold *th*, the instance is labeled as class positive (+) if score >= *th*, else it is labeled negative (-)



1. Algorithm 2 is tested on the following two datasets with similar characteristics of positive and negative classes as in the training set -  
     
   a. 1000 positives, 100 negatives  
   b. 1000 positives, 1000 negatives  
     
   For each dataset, class predictions are generated using three thresholds on the score 2 - 3, 5 and 7. Mark these points on the ROC curve generated in part 1. For each of the two datasets, report the expected precision, recall, TPR, FPR and F-measure for the class predictions using the 3 different thresholds. Also, choose the best threshold in each case based on the computed F-measure.

Case a: 1000 positives, 100 negatives

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Precision | Recall | TPR | FPR | F-measure |
| Threshold =3 | 0.9211 | 0.875 | 0.875 | 0.75 | 0.8974 |
| Threshold =5 | 0.9524 | 0.75 | 0.75 | 0.375 | 0.8392 |
| Threshold =7 | 0.9091 | 0.125 | 0.125 | 0.125 | 0.2198 |

Best threshold =3

Case b: 1000 positives, 1000 negatives

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Precision | Recall | TPR | FPR | F-measure |
| Threshold =3 | 0.5385 | 0.875 | 0.875 | 0.75 | 0.6667 |
| Threshold =5 | 0.6667 | 0.75 | 0.75 | 0.375 | 0.7059 |
| Threshold =7 | 0.5 | 0.125 | 0.125 | 0.125 | 0.2 |

Best threshold =5

1. Based on the ROC curves plotted in part 1, choose the algorithm that is expected to perform better on a test data set with similar data characteristics. Justify.  
     
   Algorithm 1 is expected to perform better than Algorithm 2 because its ROC curve is never below the ROC curve for algorithm 2 for all values of FPR. i.e for any given value of FPR, its TPR is always greater than or equal to that for algorithm 2.
2. Consider a data set with instances belonging to one of two classes - positive(+) and negative(-). A classifier was built using a training set consisting of an equal number of positive and negative instances. Among the training instances, the classifier correctly classifies a fraction m of the positive class and a fraction n of the negative class.

The trained classifier is now tested on two data sets. Both have similar data characteristics as the training set. The first data set has 1000 positive and 1000 negative instances. The second data set has 100 positive and 1000 negative instances.

1. Draw the expected confusion matrix summarizing the ***expected*** classifier performance on the two data sets.

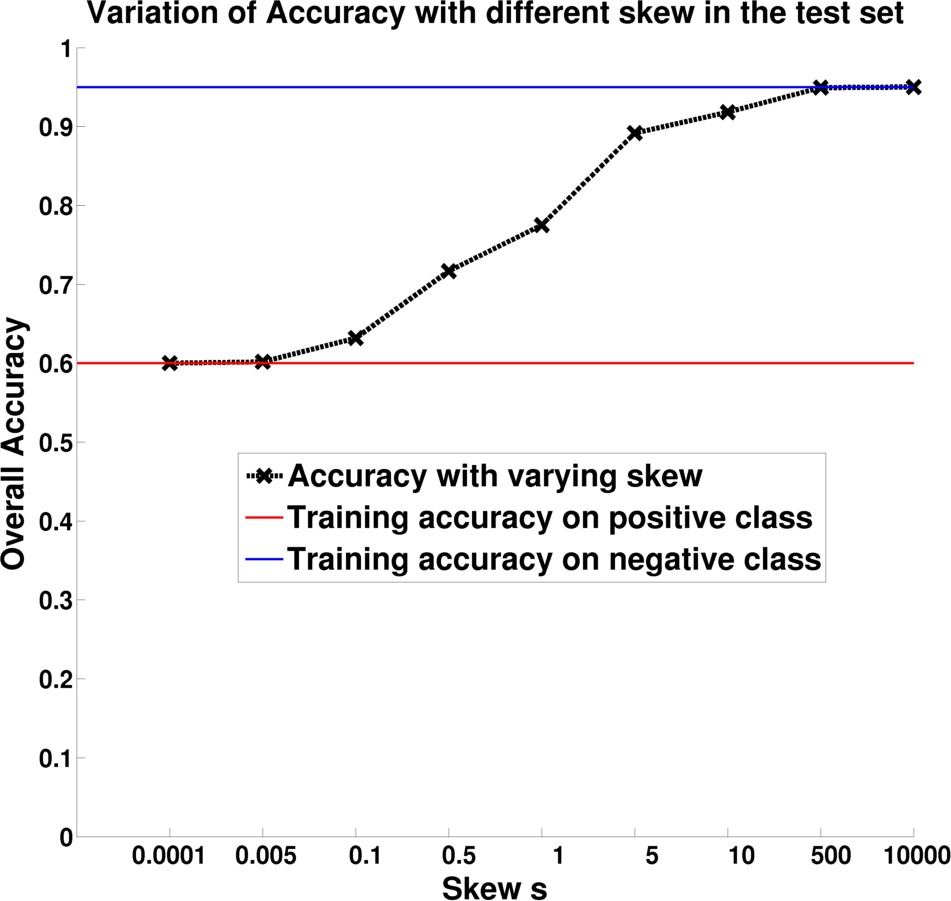
|  |  |  |
| --- | --- | --- |
|  | Algorithm Output = (+) | Algorithm Output = (-) |
| True Label = (+) | 1000m | 1000(1-m) |
| True label = (-) | 1000(1-n) | 1000n |

|  |  |  |
| --- | --- | --- |
|  | Algorithm Output = (+) | Algorithm Output = (-) |
| True Label = (+) | 100m | 100(1-m) |
| True label = (-) | 1000(1-n) | 1000n |

1. What is the accuracy of the classifier on the training set? Compute the precision, TPR and FPR for the two test data sets using the confusion matrix from part A. Also report the accuracy of the classifier on both data sets.  
     
   Training accuracy = (m+n)/2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | TPR | FPR | Accuracy |
| Data set 1 | m/(1-n) | m | 1-n | (m+n)/2 |
| Data set 2 | m/(10\*(1-n)) | m | 1-n | (m+10n)/11 |

1. i. If the skew in the test data - the ratio of the number of positive instances to the number of negative instances, is s, what is the accuracy of the algorithm on this data set? Express your answer in terms of s, m, n.  
     
   ii. In the expression for overall accuracy obtained from part (i), fix the values of m,n at m = 0.6, n = 0.95. Now compute the accuracy for the following values of s - 0.0001, 0.005, 0.1, 0.5, 1, 5, 10, 500, 10000. Plot the obtained values keeping accuracy on the vertical axis and skew on the horizontal axis. Also, plot horizontal lines corresponding to accuracy = m and accuracy = n on this graph for reference.  
     
   iii. What does the expression in (i) evaluate to if s is very large (>>1)? And when s is very small (<<1)?   
   Accuracy(m,n,s) = (m + s\*n)/(1+s),   
   If s >>1, n  
   If s<<1, m



In the scenario where the class imbalance is pretty high (say, s>500 for part C), how are precision and recall better metrics in comparison to overall accuracy? What information does precision capture that recall doesn’t?

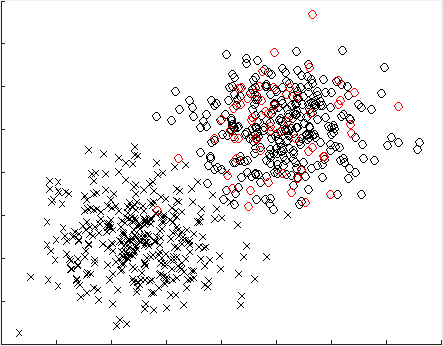
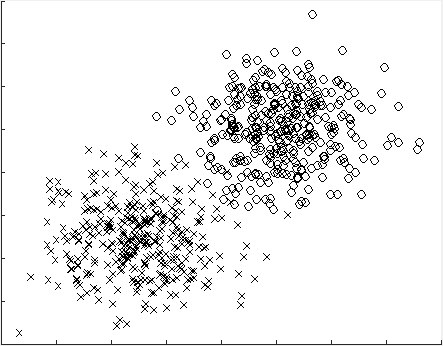
When the skew gets high, i.e., the total number of negatives are much more than the number of positives, the overall accuracy of the classifier is mainly dominated by the accuracy of the classifier on the majority negative class. From the curve in part C, we can see that the accuracy is almost constant after the skew increases beyond a certain point. If a classifier does well on the majority negative class but does poorly on the positive class, the overall accuracy wouldn’t be able to capture this.  
Precision is important to consider because it keeps track of the number of false positives of the algorithm and how significant are they in comparison to the true positives.  If your algorithm flags 1000 instances as positive, but only 100 of them are truly positives, its performance is not good even if it captures all the truly positive instances, thereby giving it a high recall.

1. Consider the framework for Rare class Prediction in the absence of True labels (RAPT). Stage 1 of the RAPT framework enables one to train a classifier using imperfectly labeled samples that, under some assumptions, is almost as effective as the one trained using expert-annotated samples (ground truth). The key assumption required therein is that the imperfect labels are conditionally independent of the features given the true labels.

See the paper, Varun Mithal, Guruprasad Nayak, Ankush Khandelwal, Vipin Kumar, Nikunj C. Oza, Ramakrishna R. Nemani: RAPT: Rare Class Prediction in Absence of True Labels. IEEE Trans. Knowl. Data Eng. 29(11): 2484-2497 (2017).

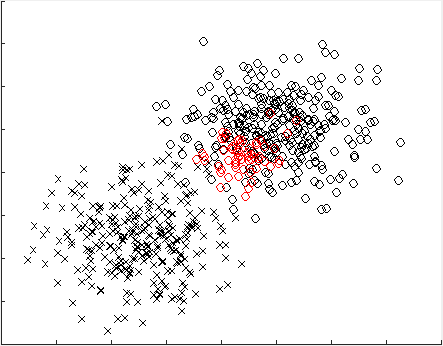
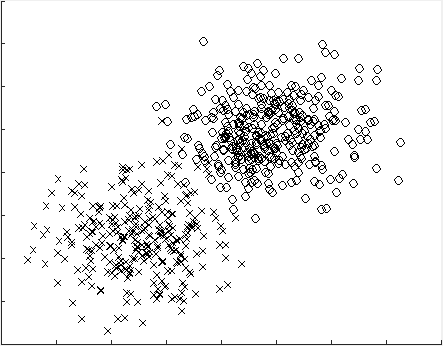
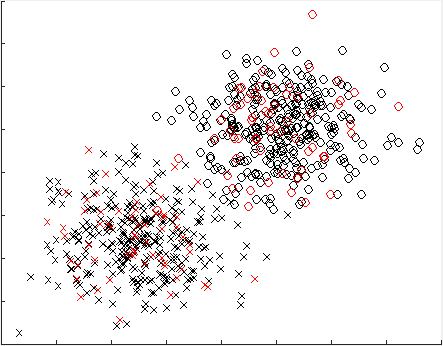
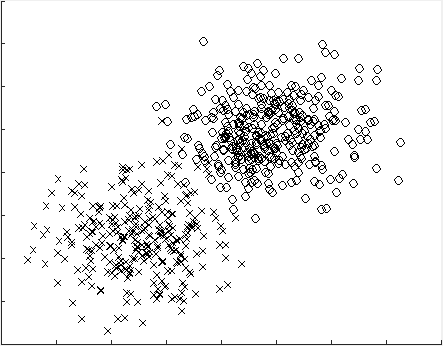
There are three pairs of figures below, A, B, and C. The figure on the left shows the true labels, with circles representing the negative class and crosses represent the positive class. The figure on the right is the same except that imperfect label has marked some of these instances incorrectly. These instances are colored red in the figure on the right.

For each of these data sets, indicate whether stage 1 of RAPT is expected to learn the separating hyperplane given the imperfect labels for training. Provide a short description in each case, explaining why you think conditional independence would/would not hold.



Data set A - True labels Data set A – Imperfect label

Data set B – True labels Data set B – Imperfect labels



Data set C – True labels Data set C – Imperfect labels

Figure A:

The conditional independence assumption holds here. There is no error in the positive class and the errors in the negative class are distributed uniformly over the span of the negative class without being biased to any specific region.   
RAPT should be able to learn the right classifier in this data set.

Figure B:

The conditional independence assumption holds here. The errors in both classes are distributed uniformly over the span of the respective class without being biased to any specific region.

RAPT should be able to learn the right classifier in this data set.

Figure C:

The conditional independence assumption is not true in this case. The errors in the negative class are biased towards a small region inside the span of the circles, rather than having the same density over the span of the negative class.

RAPT would not be able to learn the right classifier in this data set.

# **Comparison of Classification Algorithms**

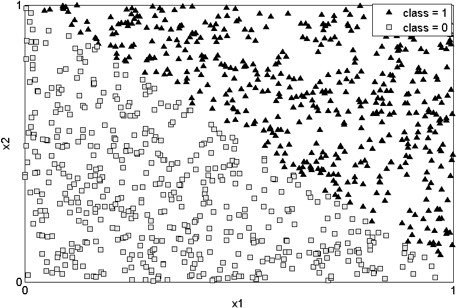
1. In each of the classification scenarios listed below, you are given a set of classifiers and a description of the classification scenario. For each scenario, state the choice of the classifier that is best suited for the dataset along with a brief explanation supporting your answer.
   1. Scenario: Data contains some missing values for certain attributes in the training and test data. Classifiers available: ANN, Naïve Bayes, RIPPER, KNN

Naïve Bayes is best suited for handling missing values in training and test data, as it can learn the conditional probabilities by ignoring missing values in the training data, and compute the product of conditional probabilities for a test instance using only those attributes that have non- missing values. ANN cannot handle missing values in the test data, as the trained ANN model cannot be applied with missing values in certain attributes. Since RIPPER involves using an ordered rule set and the rules are not mutually exclusive, the presence of missing values in the test set can lead to the firing of incorrect rules. KNN cannot handle missing values in the test data, as the distance function cannot be defined in the presence of missing values in certain attributes.

* 1. Scenario: Many of the attributes are irrelevant (contain no information about the class). Classifiers available: KNN, ANN, Decision Trees

ANN can handle the presence of irrelevant attributes. ANN can learn zero coefficients for irrelevant attributes, thus eliminating their effect on the classification performance.

Decision Trees can ignore irrelevant attributes by choosing other relevant attributes for splitting and decision tree construction, which provide higher reduction in impurity as compared to the irrelevant attributes. However, in some cases, the decision boundary is jointly determined by multiple attributes (e.g., see figure below) and none of the relevant attributes could individually obtain significant information gain compared to irrelevant attributes. As a result, the decision tree learnt includes noisy attributes and miserably fails on testing data.



KNN does not perform well in the presence of irrelevant attributes, which impact the distance function, leading to poor classification performance.

* 1. Scenario: Dataset contains attributes that are not discriminative by themselves but are discriminative in combination. Classifiers available: KNN, Naïve Bayes, Decision Trees (taking single attribute at a time)

KNN can handle the presence of interactions among attributes that are not discriminative by themselves, but are discriminative in combination, as KNN uses a distance function that simultaneously considers the effect of all attributes in the dataset. Naïve Bayes assumes conditional independence among the attributes and hence is not suited for handling interactions among attributes. Since Decision Trees consider a single attribute at a time for splitting a given node, it does not take into account the discriminative power of a combination of attributes, and hence cannot handle interactions among attributes

1. In each of the classification scenarios listed below, you are given a set of classifiers and a description of the classification scenario. For each scenario, state the choice of the classifier that is best suited for the dataset along with a brief explanation supporting your answer.

a) Scenario: Data contains some missing values for certain attributes in the training and test data.

Classifiers available: Naïve Bayes, RIPPER, KNN

Naïve Bayes is best suited for handling missing values in training and test data, as it can learn the conditional probabilities by ignoring missing values in the training data and compute the product of conditional probabilities for a test instance using only those attributes that have non-missing values. Since RIPPER may involve using an ordered rule set and the rules are not mutually exclusive, the presence of missing values in the test set can lead to the firing of incorrect rules. KNN cannot handle missing values in the test data well, as the distance function is not well-defined in the presence of missing values in certain attributes.

b) Scenario: Many of the attributes are irrelevant (contain no information about the class). Classifiers available:

KNN, Decision Trees

Decision Trees can ignore irrelevant attributes by choosing other relevant attributes for splitting and decision tree construction, which provide higher reduction in impurity as compared to the irrelevant attributes.

However, in some cases, the decision boundary is jointly determined by multiple attributes (e.g., see figure below) and none of the relevant attributes could individually obtain significant information gain compared to irrelevant attributes. As a result, the decision tree learnt includes noisy attributes and miserably fails on testing data.

KNN does not perform well in the presence of irrelevant attributes, which impact the distance function, leading to poor classification performance.

c) Scenario: Dataset contains attributes that are not discriminative by themselves but are discriminative in combination. Classifiers available: KNN, Naïve Bayes, Decision Trees (taking single attribute at a time)

KNN can handle the presence of interactions among attributes that are not discriminative by themselves, but are discriminative in combination, as KNN uses a distance function that simultaneously considers the effect of all attributes in the dataset. Since Decision Trees consider a single attribute at a time for splitting a given node, it does not take into account the discriminative power of a combination of attributes, and hence cannot handle interactions among attributes. Similarly, RIPPER only considers one attribute at a time when adding a condition to a rule and thus, cannot handle interactions among attributes.

1. In each of the classification scenarios listed below, you are given a set of classifiers and a description of the classification scenario. For each scenario, state the choice of one classifier that is best suited and another one that is worst suited. Give a brief explanation to support your answer.
2. **Scenario:** The data has mixed attributes (some are continuous, some are categorical) and some attributes are redundant (duplicate copies of other attributes).

**Classifiers for comparison:** kNN, Decision Trees, Naïve Bayes, multi-layer ANN, linear SVM

Best suited: Decision Trees

Explanation: Decision trees can handle attributes of varying types (categorical and continuous) for constructing attribute test conditions. Further, the presence of redundant attributes will have no impact on the classification performance, as only one of them will be chosen for splitting at an internal node in the tree. Naïve Bayes can also handle mixed attributes but can be impacted by redundant attributes, since all redundant attributes will have a contribution to the joint conditional probabilities. ANN and SVM may not be straightforward to use in the presence of categorical attributes and may require additional preprocessing such as transforming categorical attributes to symmetric or asymmetric binary attributes.

Worst suited: kNN

Explanation: kNN depends on the choice of a suitable proximity measure, which is impacted both by the presence of mixed attributes and the presence of redundant attributes. It is difficult to define proximity measures when different attributes have different scales and properties. Also, redundant attributes may have larger weight on the overall proximity measure and hence may produce different classification results as compared to the case where there are no redundant attributes.

1. **Scenario:** Attributes in the data are not discriminative by themselves (when considered individually) but arediscriminative when used in non-linear combinations.

**Classifiers for comparison:** Naïve Bayes, linear SVM, logistic regression, kNN

Best suited: kNN

Explanation: kNN can handle interactions among attributes and can also work with non-linearly separable data. Linear SVM and logistic regression can also handle interacting attributes but cannot handle non-linear decision boundaries.

Worst suited: Naïve Bayes

Explanation: Naïve Bayes cannot handle interacting attributes due to the conditional independence assumption, and hence will show worst performance.

1. In each of the classification scenarios listed below, you are given a set of classifiers and a description of the classification scenario. For each scenario, state the choice of one classifier that is best suited and another one that is worst suited. Give a brief explanation to support your answer.
2. **Scenario:** The data has mixed attributes (some are continuous, some are categorical) and some attributes are redundant (duplicate copies of other attributes).

**Classifiers for comparison:** kNN, Decision Trees, Naïve Bayes

Best suited: Decision Trees

Explanation: Decision trees can handle attributes of varying types (categorical and continuous) for constructing attribute test conditions. Further, the presence of redundant attributes will have no impact on the classification performance, as only one of them will be chosen for splitting at an internal node in the tree. Naïve Bayes can also handle mixed attributes but can be impacted by redundant attributes, since all redundant attributes will have a contribution to the joint conditional probabilities.

Worst suited: kNN

Explanation: kNN depends on the choice of a suitable proximity measure, which is impacted both by the presence of mixed attributes and the presence of redundant attributes. It is difficult to define proximity measures when different attributes have different scales and properties. Also, redundant attributes may have larger weight on the overall proximity measure and hence may produce different classification results as compared to the case where there are no redundant attributes.

1. **Scenario:** Attributes in the data are not discriminative by themselves (when considered individually) but arediscriminative when used in combinations. Data also contains noisy and irrelevant attributes.

**Classifiers for comparison:** Bayesian networks, decision trees, kNN

Best suited: Bayesian Networks can handle interactions among attributes. They can also work with noisy and irrelevant attributes. kNN can also handle variable interactions but it can be affected by noisy/irrelevant attributes.

Worst suited: Decision Trees. They cannot handle variable interactions and when this is combined with irrelevant attributes, they are susceptible to overfitting.

1. **Scenario:** Data contains missing values, both during training and testing.

**Classifiers for comparison:** Naïve Bayes, decision trees, kNN

Best suited: Naïve Bayes can handle missing values both during training and testing. Decision trees can handle missing values during training and testing to some extent, but it requires correction mechanisms like the probabilistic split method.

Worst suited: kNN has to discard an instance if it contains missing values either during training or testing.

1. For the following scenario, State one major strength and one major weakness of Decision Trees

**Scenario:** Some attributes in the data are not discriminative by themselves (when considered individually) but are discriminative when used in combinations. Data also contains some redundant (duplicate) and irrelevant attributes.

Strengths: Decision tree construction is not impacted by duplicate attributes (e.g., once a binary attribute is selected for splitting, then its duplicate will not be selected). Irrelevant attributes will have low values of information gain, and thus will not be preferred over other more discriminating attributes.

Weakness: In case of variable interactions (i.e., variables are not discriminating by themselves but together they can be discriminating), it is possible for irrelevant attributes to be selected before such potentially informative attributes. This can lead to large and overfitted trees, especially if there are a large number of irrelevant attributes.

1. Consider the following classification methods: Decision Trees, RIPPER, Support Vector Machines, Naïve Bayes, k-Nearest Neighbor and Artificial Neural Network. For each of the following scenarios, state the choice of a classifier that is well suited and another one that is poorly suited. Give a brief explanation to support your answer.
2. Many attributes are redundant (i.e., they are duplicate versions of other attributes).

Well Suited: DT, RIPPER, SVM, ANN

Explanation: DT and RIPPER can handle redundant attributes as the redundant attributes are not selected for splitting or rule construction. SVM, ANN would also be able to handle redundant attributes.

Poorly Suited: k-NN, NB

Explanation: The distance measure of k-NN can start giving more weight to the redundant attributes due to their multiplicity, which can potentially create issues. Naïve Bayes would have a difficult time working with redundant attributes due to variable interaction. Each redundant attribute will affect the posterior probability.

1. The number of attributes is large and many of them are irrelevant or uninformative (i.e., they provide no discriminative power individually or in combination with other attributes).

Well Suited: NB, DT, RIPPER, SVM, ANN

Explanation: NB can handle irrelevant attributes because the conditional probability of the irrelevant attributes would be close to 0.5, leading to no effect in the posterior probabilities of both classes. DT and RIPPER can show good performance even in the presence of irrelevant attributes given that there are some discriminative attributes in the data that are not interacting with each other and the model is not prone to overfitting. SVM and ANN will also do well as discriminative attributes dominate their models.

Poorly Suited: kNN

Explanation: It treats both informative and non-informative attributes similarly and hence distance values are affected by irrelevant attributes.

1. There are attributes that are not discriminative individually, but their combination provides the information to separate the classes.

Well Suited: k-NN, SVM, ANN

Explanation: k-NN can handle interactions as the entire feature space is considered together while making decisions on the predicted class. SVM and ANN can also handle variable interactions since they can construct linear as well as non-linear decision boundaries.

Poorly Suited: NB, DT, RIPPER

Explanation: NB cannot handle interactions between attributes, as it requires conditional independence assumption between attributes. DT and RIPPER also look at one variable at a time while making decisions on the predicted class, and hence will have problems with variable interaction.

1. Computation time for model building is to be minimized.

Well Suited: kNN

Explanation: There is practically no model building in kNN.

Poorly Suited: ANN

Explanation: Weights for each edge in network has to be learned in an iterative fashion for ANN, and it might take a long time to converge. SVM also has high model complexity (can be a second choice). DT, RIPPER and NB are quite computationally efficient as compared to ANN and SVM.

1. Test records contain missing values.

Well Suited: NB

Explanation: Posterior probability can be calculated in the presence of incomplete information. DT can also handle missing values in test records, but not as efficiently as NB (can be considered as the second choice).

Poorly Suited: SVM, ANN, k-NN, RIPPER

Explanation: SVM, ANN, and RIPPER require all the attributes to be present in the test data. Otherwise, no decision can be made. RIPPER can ignore the triggering of a rule with missing attributes and proceed to the next rule in the ordered list, but it might end up triggering an incorrect rule and hence can suffer from high testing errors.

1. If you had to choose between the naive Bayes and k-nearest neighbor classifiers, which would you prefer for a classification problem where there are numerous missing values in the training and test data sets? Indicate your choice of classifier and briefly explain why the other one may not work so well?

The naive Bayes classifier is a much better choice in this case since the impact of missing values on the calculation of probabilities is little as long as enough values are available for each attribute. On the other hand, the similarity calculation step of **k**-NN will be significantly impacted depending on which attribute values are missing between the examples whose similarity is being computed. For instance, consider three data points **A**, **B** and **C**, where the values for attributes 1 and 2 are missing for **B** and **C,** respectively. Due to these missing values, the similarity function may determine that **B** is closer than **C** to **A**, even though the reality may be the converse. Also, the similarity between **B** and **C** would be undefined in this case. Due to these effects of missing values on the similarity function, missing values in data are expected to have an adverse effect on the performance of the **k**-nearest neighbor classifier.

1. Consider the problem of predicting whether a person is a good credit risk given the following attributes: hair color, income, weight, time in the current job, marital status, height, age, and birth month. If you had to choose between Ripper and a k-nearest neighbor classifier, which would you prefer? Indicate your choice of classifier and briefly explain why the other one may not work so well?

Ripper is preferred over KNN in this problem since the variables appear to be of

varying relevance. KNN’s performance can be adversely impacted by irrelevant attributes, as

they can unduly influence the similarity function. Ripper, like most other rule-based models,

as well as decision tree classifiers, performs variable selection using measures such as the Gini

index that, although not perfect, allow irrelevant variables to be discarded. The rules generated by Ripper are also more interpretable, unlike the KNN classifier.

1. A realtor is studying housing values in the suburbs of Minneapolis and has given you a dataset with the following attributes: crime rate in the neighborhood, proximity to Mississippi river, number of rooms per dwelling, age of unit, distance to Minneapolis and Saint Paul Downtown, distance to shopping malls. The target variable is the cost of the house (with values high and low). Given this scenario, indicate the choice of classifier for each of the following questions and give a brief explanation.
2. If the realtor wants a model that not only performs well but is also easy to interpret, which one would you choose between SVM, Decision Trees and kNN?

SVM and Decision Trees may have somewhat similar performance, but Decision Trees are easier to interpret. kNN may not perform well due to the mixed types of attributes in the dataset, which can make it difficult to devise the right proximity measure.

1. If you had to choose between RIPPER and Decision Trees, which one would you prefer for a classification problem where there are missing values in the training and test data?

Both Decision Trees and RIPPER can handle missing values in the training data since they can only consider the non-missing values while considering discriminating (splitting) attributes. However, Decision Trees can handle a missing value in the test data by branching along both children at a node requiring the missing value and using the impurities at the multiple leaves reached to decide on the predicted class. On the other hand, a missing attribute might not allow the correct rule in RIPPER to trigger and an incorrect rule can get triggered eventually since decisions at subsequent rules depend on decisions at earlier rules due to the ordering of the rules.

1. If you had to choose between RIPPER and KNN, which one would you prefer if it is known that there are very few houses that have high cost?

RIPPER. Both RIPPER and kNN are good in handling class imbalance, but the attributes are very different as they have different scales. Thus kNN will not perform well.

1. Suppose you have a dataset with many different types of variables (categorical, continuous, binary, etc.) and many missing values and a number of redundant attributes. Also, the data contains a large number of attributes, some of which are discriminative. You can select one of the following classifiers to run on this dataset: decision tree, SVM, ANN. Which one will you choose and why? Also, explain why the other two wouldn’t work as well.

Answer: Decision Tree.

Decision tree building involves ranking the attributes using measures such as information gain, and if two attributes are redundant, only one of them will be used in building the tree. SVMs and ANNs can also handle redundant attributes, but it will be difficult to apply them in the presence of categorical attributes (unless the categorical attributes are binarized into a number of binary attributes). On the other hand, a decision tree can naturally handle different types of variables.

1. Given three choices: Decision Trees, Naïve Bayes and Artificial Neural Networks (ANN) give the best and worst choice for the following scenarios. Provide a simple explanation. While answering this question, you are allowed to make additional reasonable assumptions, but you must clearly state them.
2. Dataset has many redundant attributes.

Best Choice: Decision Tree or ANN

Worst Choice: Naïve Bayes

Explanation: DT will select only one of the redundant attributes and ignore the others. ANN will assign lower weights to redundant attributes.

Naïve Bayes’ class-conditional-independence assumption is invalid here

1. In the test dataset, only some of the attribute values are known.

Best Choice: Naïve Bayes

Worst Choice: ANN

Explanation: Naïve Bayes can ignore the missing values in the posterior probability distribution. ANN requires the values of all attributes to generate classification results.

1. Consider a scenario where you are supposed to determine if a person has heart disease or not based on the following attributes: blood pressure, body weight, age, ethnicity, height, number of cigarettes consumed in a week, and type of job (which can take four values: business, healthcare, engineering, or education).
2. State one strength and one weakness of kNN for this task?

**Strength:** kNN can handle variable interactions and learn more complex boundaries.

**Weakness:**Unable to identify which features are relevant for discrimination and their relative importance.

Potentially high computational cost during the testing phase.

1. State one strength and one weakness of decision trees for this task?

**Strength:**

Decision Trees are highly interpretable, can work with both categorical and continuous attributes, and performs automatic variable selection.

**Weakness:**

Uses only one attribute at a time for splitting, which could lead to problems in case there are interactions between attributes and irrelevant attributes are present as well.

1. What aspects of this problem might lead you to choose Naïve Bayes over SVM?

There could be potential missing values in the data. For example, ethnicity of some people could be missing. In such scenarios, Naïve Bayes would be preferred over SVM because it could easily handle missing values in the data.

1. What aspects of this problem might lead you to choose RIPPER over Decision Trees?

It could be expected that the number of people with heart disease is typically much lower compared to healthy people. RIPPER can handle such a skewed class distribution more effectively than Decision Trees due to its ability to focus on the rare class

1. Consider a scenario where you are supposed to build a classification model to determine if a person has heart diseases or not based on the following attributes: blood pressure, body weight, age, number of cigarettes consumed in a week, and type of job (which can take four values: business, healthcare, engineering, or education).
2. What characteristic(s) of this problem will make you prefer decision trees over k-NN?

The attributes are of mixed types, which would make it more difficult to define a distance or similarity measure. Also, since this is a medical scenario, the understandability of the model is important, and a decision tree is more interpretable than kNN.

1. What characteristic(s) of this problem will make you prefer Naïve Bayes over decision trees?

Naïve Bayes will give us the posterior probability for both the case where the patient has heart disease and the probability that they don’t. This will give us more information than just a binary classification, which is what a decision tree gives. Also, naïve Bayes is better at handling missing values than decision tree and medical data may well have missing values.

1. If you had to use SVM for classification, which transformation, if any, would you require to pre-process the data before learning the classification model?

Convert the nominal attribute (type of job) into 4 asymmetric binary variables, each indicating the presence of a particular type of job.

1. Consider the problem of predicting the risk of liver injury (high risk or low risk) for a patient, given the following attributes: age, weight, level of various liver enzymes, ethnicity, alcohol dependence, and type of jobs (which can take four values: business, healthcare, engineering, or education).
2. State one strength and weakness of naïve Bayes for this task?

Strength: naïve Bayes can handle the presence of irrelevant attributes (ethnicity, job type) Weakness: naïve Bayes assumption on conditional independence will fail in the presence of interacting attributes (alcohol dependence and enzyme level)

Note: Other answers that will be given full credits:

**Strength**:

- Inexpensive to build, especially because this data has many attributes (high dimensional)

Answers that will be given partial credits:

Any valid Naïve Bayes strength and weakness that are not presented in the context of the problem given. Such as:

**Strength**:

- Naïve Bayes handles missing attribute values well.

- Naïve Bayes is a probabilistic model

1. State one strength and weakness of kNN for this task?

**Strength**: kNN can handle interaction between attributes (e.g: alcohol dependence might be related to level of liver enzymes) and can handle arbitrary attribute shapes.

**Weakness**: kNN cannot handle the presence of irrelevant attributes (e.g: ethnicity and types of job) since it can add bias to the proximity measurement.

Note: Other answers that will be given full credits:

**Weakness**:

- Attributes are of different types (i.e: continuous, categorical), we will need to standardize to make sure no attribute dominates the distance measure.

Answers that will be given partial credits:

Any valid kNN strength and weakness that are presented in the context of the problem given. Such as:

**Strength**:

- kNN is model free. When new instance come, we classify based on the closest instances

- kNN can represent complex decision boundaries

**Weakness**:

- kNN cannot handle missing attribute values.

1. If you had to choose between RIPPER and decision tree, which one would you choose? Briefly justify your choice

RIPPER. It could be expected that the number of people with a high risk of liver injury is typically much lower than the low risks. RIPPER can handle such a skewed class distribution more effectively than Decision Trees due to its ability to focus on the rare class

1. Consider a data set with four binary attributes X1, X2, X3 and X4. The attribute X4 takes exactly the same value as X3 for each record, i.e., X4 is equal to X3. In each of the following three scenarios, find whether the decision boundary learnt by the two models would be similar, otherwise find which of the two models would perform better. Provide a brief justification for each.

(i) We build two decision trees:

a. T1, that is learnt using all the four attributes

b. T2, that is learnt using only three attributes X1, X2, and X3.

The performance of T1 and T2 would exactly be the same as the inclusion of attribute X4, which is identical to X3 would not alter the choice of splitting attribute at each node.

(ii) We build two Naïve Bayes models:

a. B1 that is learnt using all the four attributes.

b. B2 that is learnt using the three attributes X1, X2, and X3.

Since X4 is the duplicate of X3, the assumption of class conditional independence among the attributes is violated for B1 and therefore, the decision boundary learnt in B1 would not be the same as that of B2. B1 would be more influenced by the class distribution along attribute X3, which could improve or degrade the performance of B1 as compared to B2, depending upon how well the two classes are separated along the attribute X3. The points for which attributes X1 and X2 are of higher relevance to obtain correct class labels are more likely to be misclassified by B1.

(iii) We build two KNN models:

a. M1 that is learnt using all the four attributes.

b. M2 that is learnt using the three attributes X1, X2, and X3.

Since X4 is duplicate of X3, model M1 will effectively give double weight to X3 compared to the other two attributes X1 and X2, which could improve or degrade the performance of M1 as compared to M2, depending upon the relative relevance of the attribute X3 compared to the other two attributes X1 and X2. The points for which attributes X1 and X2 are of higher relevance to obtain correct class labels are more likely to be misclassified by M1.

(iv) We build two ANN models:

a. M1 that is learnt using all the four attributes.

b. M2 that is learnt using the three attributes X1, X2, and X3.

The performance of both M1 and M2 is expected to be the same, as the neural network M2 would assign equal weights to X3 and X4, so that wt(X3) + wt(X4) would be same as wt(X3) in M1, while the weight assigned to X1 and X2 would remain unchanged.

1. Consider a data set with four binary attributes X1, X2, X3 and X4. The attribute X4 takes exactly the same value as X3 for each record, i.e., X4 is equal to X3. We build two decision trees: one using all four attributes X1, X2, X3 and X4 and the other using only three attributes X1, X2 and X3.

(a) Do you expect the predictions for a new test instance using the two decision trees to be the same or different? Briefly explain.

Same. The last attribute will never be used for splitting.

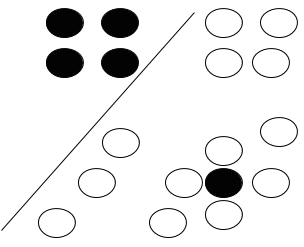
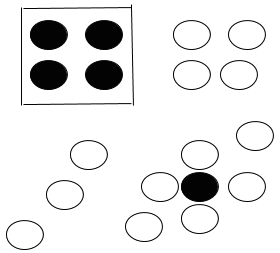
(b) Do you expect the decision tree built using all 4 attributes to be better or worse than the decision tree built using 3 attributes. Briefly explain.

Same. Since predictions are the same, there is no change in accuracy.

(c) We now build two naive Bayes models: one using all four attributes X1, X2, X3 and X4 and the other using only three attributes X1, X2 and X3. Do you expect the predictions for a new test instance using the two naive Bayes models to be same or different? Briefly explain.

Different. Additional probability term corresponding to X4.

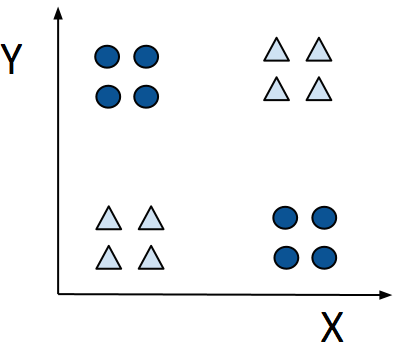
1. The figure below shows a two-dimensional training data with two target classes (represented as dark and light circles) and the classification boundaries produced by one of the following classifiers: (1) Linear Support Vector Machine, (2) 1-­‐ Nearest Neighbor classifier and (3) RIPPER. For each of the two scenarios, identify the classifier(s), which can learn the given boundary with a brief justification.



(a) (b)

|  |  |
| --- | --- |
| Scenarios | Classifier |
| (a) | RIPPER. Linear Support Vector Machine can’t learn non‐linear  boundaries. 1-NN would also learn boundary around the isolated circle, whereas RIPPER with pruning can  learn the boundary shown in scenario (a). |
| (b) | Linear Support Vector Machine, since the boundary learnt is linear. And the other two cannot learn boundaries like it. |

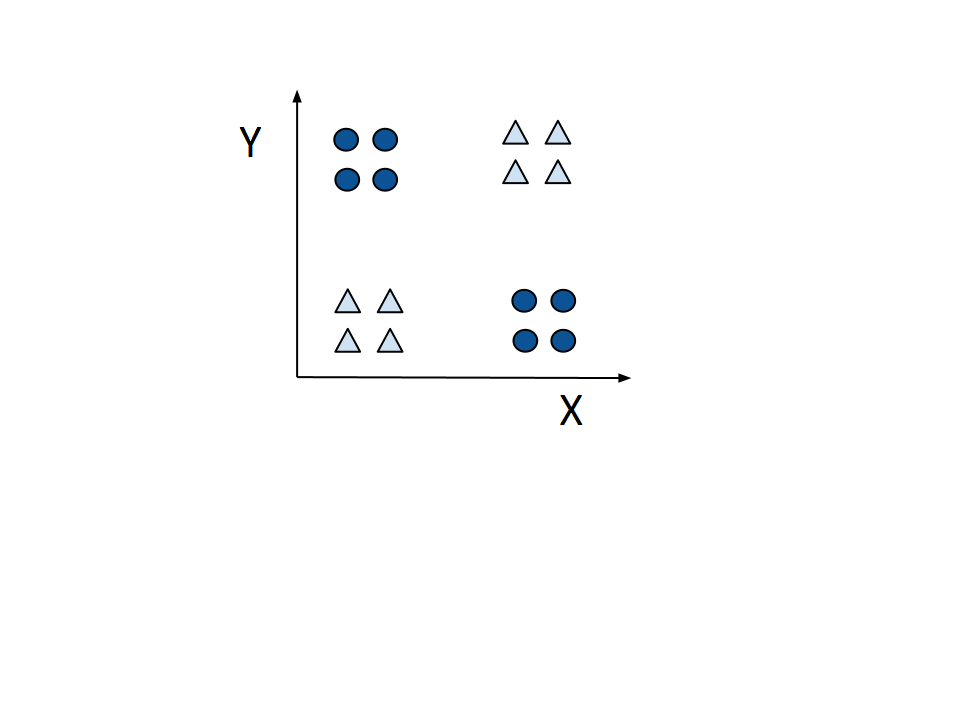
1. Choosing appropriate classifiers



For the dataset in Figure above, among k-NN, Decision Trees, Naïve Bayes, and ANN, which classifier would have the worst performance? Provide a brief explanation justifying your choice.

Naïve Bayes, as the attributes are not conditionally independent. All other classifiers would be able to handle the interaction between attributes X and Y and would have better performance than Naïve Bayes.

1. The figure below shows a data set with two attributes (attribute 1 and attribute 2 ) and two target classes. If you were constrained to only use neural network and naive Bayes classifier, which one would you use? Why?



A multi-layered neural network will be able to separate the classes. Naive Bayes is a poor choice because each attribute has the same distribution of the two classes, and the two attributes are not conditionally independent of each other given the class. Therefore, NB will be randomly guessing.

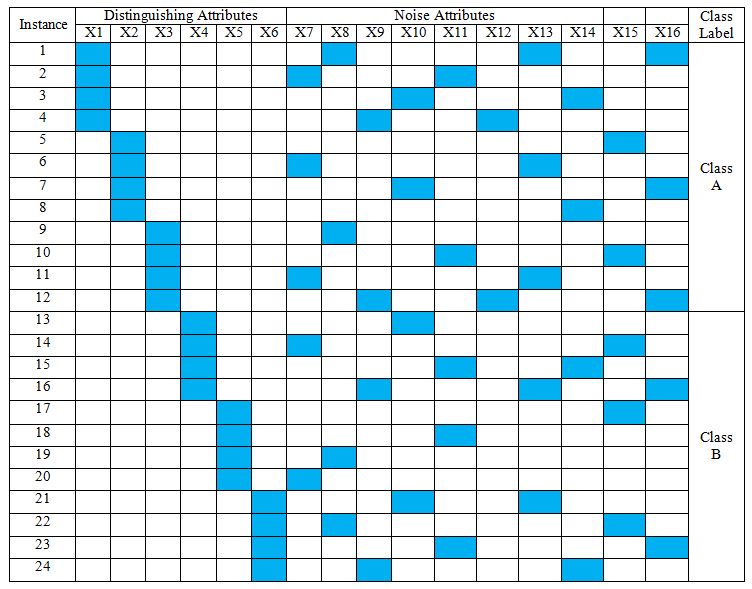
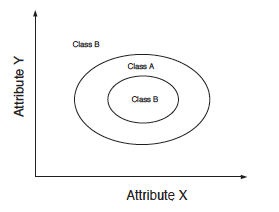
1. For the dataset in Figure below, among KNN, Naïve Bayes and ANN, which classifier would have the **worst** performance?

A picture containing drawing

Description automatically generated

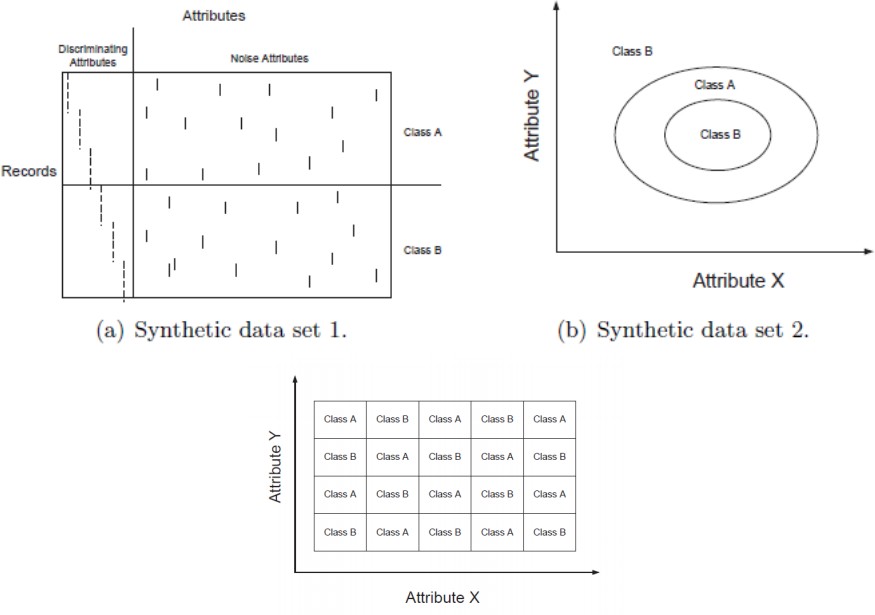
Naïve Bayes, as the attributes are not conditionally independent. K-NN and ANN can handle variable interactions

1. Given the data sets shown in Figure below, explain how the decision tree, naïve Bayes (NB), and k-nearest neighbor (k-NN) classifiers would perform on these data sets.

**(a) Synthetic data set 1. (b) Synthetic data set 2.**

1. Both decision tree and NB will do well on this data set because the distinguishing attributes have better discriminating power than noisy attributes in terms of entropy gain and conditional probability. k-NN will not do as well due to the relatively large number of noise attributes, which will adversely affect the computation of similarity between two examples.
2. k-NN will work best due to the proximity of the examples of the same class to each other. NB does not work well for this data set since the attributes that determine the class boundaries are not independent. A decision tree will have to be large in order to capture the circular decision boundaries and thus is not the ideal solution.
3. Given the data sets shown in the three subfigures below, list the ones that are suitable for each of the following classifiers, naive Bayes (NB), decision tree and k-nearest neighbor (k-NN) classifiers would perform on these data sets. (a) A binary dataset represented as a binary matrix with each row being a record and each column being an attribute. The rows are grouped into two classes A and B. For each variable, the dashed or real vertical lines indicate 1’s in the matrix and the rest are 0’s. (b & c) Each represents a dataset with two variables (x and y-axis) and two classes (A and B) of data points separated by the boundaries.



1. Synthetic data set 3

(a) On Synthetic Data set 1: Both decision tree and NB will do well on this data set because the distinguishing attributes have better discriminating power than noisy attributes in terms of entropy gain and conditional probability. k-NN will not do as well due to the relatively large number of noise attributes, which will adversely affect the computation of similarity between two examples.

(b) On Synthetic Data set 2: k-NN will work the best due to the proximity of the examples of the same class to each other. NB does not work well for this data set since the attributes that determine the class boundaries are not independent. A decision tree will have to be large in order to capture the circular decision boundaries and thus is not the ideal solution.

(c) For synthetic dataset 3: Like dataset 2, but here the decision tree could work because the boundaries are either vertical or horizontal. KNN is the best.

1. Consider the following classification methods: Decision Trees, RIPPER, Support Vector Machines (SVM), Naïve Bayes, k-Nearest Neighbor (kNN).

Which of the above methods is most appropriate in the following situations? Give a brief explanation. If there is more than one possible answer, choose the one that you consider most appropriate, and justify your answer.

1. The class distribution is skewed.

RIPPER, since it focuses on extracting rules one class at a time and starts with rare class.

1. The number of attributes is very large, and all attributes are numeric, but only a subset of them is relevant to discriminating between the classes.

SVM, since it can directly handle continuous attributes and weights the attributes to indicate their utility for discrimination.

1. Many attributes are correlated

Decision Trees, SVM, or Ripper since naïve Bayes assumes conditional independence and the distance measure will be distorted by correlated attributes for kNN.

1. Computation time for training is to be minimized.

kNN since it has no training step.

1. The data contains missing values.

Naïve Bayes. Decision tree and RIPPER can also handle this problem to a certain extent during training (but not during testing).

1. For each of the following scenarios, select the classification method from the given set of choices that is most appropriate. Give a brief explanation justifying why the choice you made is best among the given choices.
2. Computation time for training is to be minimized. (kNN, ANN, Linear SVM, Decision Trees)

kNN. Unlike kNN, all other listed methods need to be trained, which is computationally expensive.

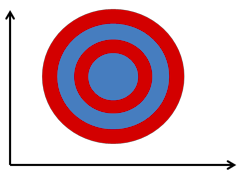
1. The number of attributes is very large and all attributes are numeric, but only a subset of them are relevant to discriminating between the classes. (Linear SVM, k­NN, Naïve Bayes, Decision Trees)

NB or DT is the best choice, but both of them have limitations in this case.

First, kNN cannot work well because it considers all attributes equally, including the irrelevant ones. Second, Linear SVM can ignore the irrelevant attributes by weighting them as zero, so high dimensionality will not impact its performance; however, the classes are not guaranteed to be linearly separable.

NB and DT cannot handle variable interactions. But at least they are both robust to the irrelevant attributes.

1. The figure below shows a dataset of two classes whose points are shown in blue and red color on a two-dimensional Cartesian coordinate axis. Among KNN, Naïve Bayes and Decision Trees, which classifier would have the best performance? Provide a brief explanation justifying why your choice is best among the three methods. (Concept: Which classifier can learn complex boundaries when there is an interaction between attributes?)



kNN – Since it looks only at the local neighborhood, it will be able to learn the complex boundary between the classes. NB will have poor performance due to variable interaction. DT will have to construct an overly complex tree due to the non­linearity in the decision boundaries, making it prone to overfitting.

1. Choosing appropriate classifiers



1. For the dataset in Figure above, choose which classifier would have the best performance among 1-NN, Decision Tree, and Naïve Bayes, which classifier would have the best performance? Assume that only a small, randomly selected sample of data (e.g., 10%) will be made available as labeled data for building the model. You are asked to comment on the expected performance on the remaining data. Provide a brief explanation justifying your choice.

1-NN. Since it looks only for local neighborhoods and can better handle the presence of interacting attributes (a combination of X and Y rather than X or Y). Decision tree will also do fine, although its split at the root node is not likely to be very informative.

1. Which classifier would have the worst performance? Provide a brief explanation justifying your choice.

Naïve Bayes, as the attributes are not conditionally independent. More specifically, given the class label (red or blue) of an object, the value of one of the attributes can tell a lot about the value of the other attribute.

1. Let’s now generate a new dataset that includes the 2 attributes X and Y from the original data set, and 95 exact duplicates of X. What impact will this have on the relative suitability of all three classifiers mentioned above? Which one is best? Which one is the worst? Briefly explain.

Performance of 1-NN will degrade, since we have a large number of redundant attributes which can lead to an overly biased proximity measure toward X, resulting in improper estimates of distance.

Naïve Bayes will still not do well. Besides the same reason in (i), the duplicated attributes of X will violate the naïve Bayes assumption of conditional independence, and thus further degrade the performance of naïve Bayes classifier.

Decision tree can still handle this problem. Since redundant attributes show similar gains in purity, the decision tree will just select any one of them to do the split when building the tree.

Hence decision tree will perform the best and Naïve Bayes will continue to be worst. 1-NN will perform closer to Naïve Bayes as the number of redundant attributes increases.

1. We generate another new dataset, which includes the 2 attributes X, and Y from the original data set, and 95 additional noisy attributes. The values for the noisy attributes are randomly assigned from the uniform distribution on [0, 20]. What impact will this have on the relative suitability of all three classifiers mentioned above? Briefly explain.

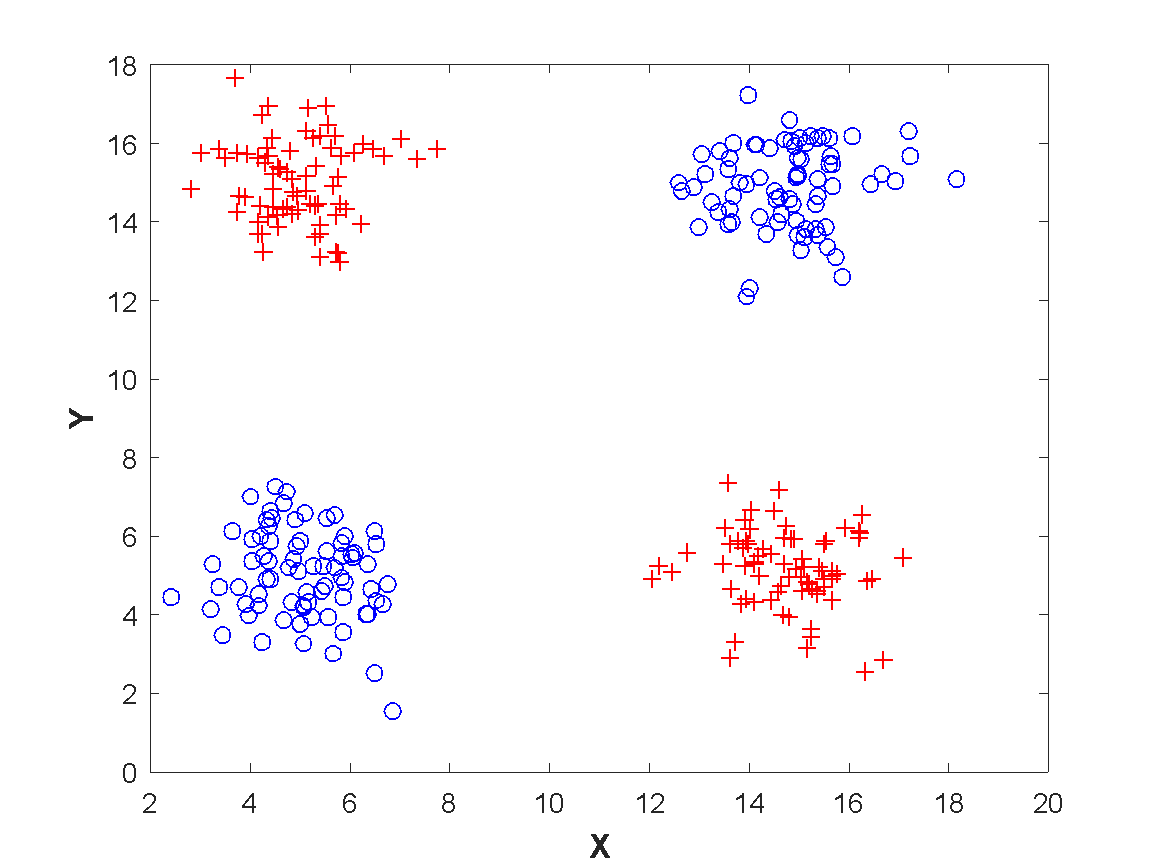
1-NN will not do well due to a relatively large number of noise attributes, which will adversely affect the computation of similarity between two examples.

Decision tree will not do well also. Since there are so many irrelevant attributes, some of them may be accidentally chosen during the tree-growing process to fit the labeled data available for training and cross validation, which will adversely affect the generalization performance of the decision tree. Note that Decision tree could have easily handled irrelevant attributes if there were fewer of them.

Naïve Bayes will have a similar performance as in (i), as it is robust to irrelevant attributes.

Thus, for this extreme case involving a very large number of irrelevant attributes, all three will have poor performance.

1. Choosing appropriate classifiers



1. For the dataset in Figure above, choose which classifier would have the best performance among 1-NN, Decision Tree, and Perceptron (first general Neural Network that tries to find a linear hyperplane separating the 2 classes), which classifier would have the best performance?  Assume that only a small, randomly selected sample of data (e.g., 10%) will be made available as labeled data for building the model. You are asked to comment on the expected performance on the remaining data.   Provide a brief explanation justifying your choice.

1-NN will perform the best. Since it looks only for local neighborhoods, it can easily handle the presence of interacting attributes (where a combination of X and Y is more discriminating than X or Y individually). Decision tree will also do fine, although its split at the root node is not likely to be very informative.

1. Which classifier would have the worst performance? Provide a brief explanation justifying your choice.

Perceptron. The classes are not linearly separable

1. Let’s now generate a new dataset that includes the 2 attributes X and Y from the original data set, and 95 exact duplicates of X. What impact will this have on the relative suitability of all three classifiers mentioned above?  Which one is best? Which one is the worst? Briefly explain.

Performance of 1-NN will degrade, since we have a large number of redundant attributes which can lead to an overly biased proximity measure toward X, resulting in improper estimates of distance.

Perceptron will still not be able to separate them, because the classes are not linearly separable, even in this new feature space.

Decision tree can still handle this problem. Since redundant attributes show similar gains in purity, the decision tree will just select any one of them to do the split when building the tree.

Hence the decision tree will perform the best and Perceptron will continue to be worst.  1-NN will perform closer to Perceptron as the number of redundant attributes increases.

1. We generate another new dataset, which includes the 2 attributes X, and Y from the original data set, and 95 additional noisy attributes. The values for the noisy attributes are randomly assigned from the uniform distribution on [0, 20]. What impact will this have on the performance of each of these 3 classifiers? Briefly explain.

1-NN will not do well due to a relatively large number of noise attributes, which will adversely affect the computation of similarity between two examples.

Decision tree will not do well also. Since there are so many irrelevant attributes, some of them may be accidentally chosen during the tree-growing process to fit the labeled data available for training and cross validation, which will adversely affect the generalization performance of the decision tree. Note that Decision tree could have easily handled irrelevant attributes if there were fewer of them.

Perceptron will have a similar performance as in (i), as the classes are not linearly separable even in this new feature space.

Thus, for this extreme case involving a very large number of irrelevant attributes, all three will have poor performance.

1. Consider the problem of predicting whether a movie is popular or not, given the following attributes: Format (DVD/Online), Movie Category (Comedy/Documentaries), Release Year, Number of world-class stars, Director, Language, Expense of Production and Length. If you have to choose between RIPPER and a k­nearest neighbor classifier, which would you prefer and why? Briefly explain why the other one may not work so well?

RIPPER is preferred over k­NN in this problem since the variables are diverse in nature and thus are expected to be of varying relevance for differentiating between the “popular” and “not popular” classes. k­NN’s performance can be adversely impacted by irrelevant attributes, as they can unduly influence the similarity function. RIPPER, like most other rule-based models, as well as decision tree classifiers, performs variable selection using measures such as the Gini index that, although not perfect, allow irrelevant variables to be discarded.

1. Imagine you are given the task to predict the educational qualification of each person using their demographic data with the following attributes: (1) Annual Income (real-valued), (2) Income Tax filed (real-valued), (3) Age (integer), (4) State of residence in the US (categorical) and (5) House Owner or not (boolean). Assume the target classes are (a) college degree and (b) no college degree. While answering this question, you are allowed to make additional reasonable assumptions, but you must clearly state them.

(i) Given the choice of Artificial Neural Network (ANN) and k-Nearest Neighbor classifier, which

one would you prefer? Mention the strength of your choice and the weakness of the other.

Your choice: ANN

Strength of your chosen classifier:

ANN can handle the presence of redundant attributes (e.g., Annual income and income tax) and it can automatically identify features that are discriminative and ignore the irrelevant (uninformative ones).

Weakness of the other classifier:

kNN cannot automatically identify features that are most relevant for class discrimination and will require a hand-crafted similarity/dissimilarity measure that can be difficult to obtain.

(ii) Given the choice of Naïve Bayes and Support Vector Machines (SVM), which classifier would

you choose? Mention the strength of your choice and the weakness of the other.

Your choice: SVM

Strength of your chosen classifier: SVM can handle irrelevant and redundant attributes.

Weakness of the other classifier: Naïve Bayes requires the attributes to be independent of each other given a class (the class-conditional independence assumption), which is not valid in the presence of interacting or redundant attributes. In addition, it is harder to use for continuous values unless a parametric form of the distribution is known.

(iii) For this part, assume that the population is highly imbalanced (i.e., the fraction that has a

college degree far exceeds the fraction that does not). Given the choice of RIPPER and decision

trees, which classifier would you choose? Mention the strength of your choice and the weakness

of the other.

Your choice: Ripper

Strength of your chosen classifier: RIPPER is better able to handle imbalance data, as it makes the large class to be the default class and learn rules for the smaller class.

Weakness of the other classifier: The measures used for selection of attributes (to split nodes) in DT can all appear to have very good values (e.g., low entropy) when one class is much larger than the other.

1. Imagine you are given the task to predict the educational qualification of each person using their demographic data with the following attributes: (1) Annual Income (real-valued), (2) Income Tax filed (real-valued), (3) Age (integer), (4) State of residence in the US (categorical), (5) Gender (categorical), (6) House Owner or not (Boolean) and (7) Height (in inches). Assume the target classes are (a) college degree and (b) without a college degree. Also, assume that the fraction of the population that has a college degree is roughly equal to the fraction that does not have a college degree.

(a) State one strength and one weakness of using KNN for this task.

KNN:

Strength:

No time taken to learn a model.

Model-free method

Arbitrary boundaries can be learnt.

It is natural to use in the task.

Weakness:

Hard to define the distance matrix (different weight and different type of attributes) or need preprocessing to convert the nominal into binary attributes.

Affected by noise.

Hard to choose K.

Computational complexity to test data is large when the size of training data is large.

(b) State one strength and one weakness of using a decision tree for this task.

DT:

Strength:

Easy to interpret.

Can deal with correlated or redundant data

Inexpensive to test a data instance

Can perform automatic variable selection

Can handle missing data in both training and testing data.

Can deal with all types of data (includes categorical, Boolean, and numerical data).

Weakness:

It is a greedy approach to attributes selection.

Finding a good split condition for a continuous variable is time consuming.

Considers only one attribute at a time and thus, it is not suited for identifying interaction among variables.

(c) State one strength and one weakness of using a support vector machine for this task.

SVM:

Strength:

Global optimal solution, with respect to its loss function (even though there is no guarantee that it will generalize well).

Can deal with nonlinear separation.

Good to handle high-dimensional data.

Weakness:

non-numeric attributes

Hard to choose parameters, such as kernel function and slack variables.

Computational complexity is large

Mistakes (from students)

Cannot deal with non-linear separation.

SVM generalizes poorly on unseen data

**(d)** Assume that the test records have missing values for many attributes. Given a choice between naive Bayes and artificial neural network (ANN), which classifier would you prefer in this scenario and why?

Naïve Bayes method is preferred since it can provide an answer with any number of missing attributes.

**(e)** For this part, assume that the target classes are (a) has a Ph.D. degree and (b) does not have a Ph.D. degree. Given a choice between a decision tree and Ripper (rule-based classifier), which classifier would you prefer for this task and why?

Since the number of people with a PhD degree is likely to be very small, Ripper is preferred due to its ability to focus on small classes.

1. Imagine you are given the task to predict the educational qualification of each person using their demographic data with the following attributes: (1) Annual Income (real-valued), (2) Income Tax filed (real-valued), (3) Age (integer), (4) State of residence in the US (categorical), (5) Gender (categorical), (6) House Owner or not (Boolean) and (7) Height (in inches). Assume the target classes are (a) college degree and (b) without a college degree. Also, assume that the fraction of the population that has a college degree is roughly equal to the fraction that does not have a college degree. State one strength and one weakness of decision trees for this task?

Strengths:

Easy to interpret.

Can deal with correlated or redundant data.

Inexpensive to test data.

Can deal with all types of data (includes categorical and numerical data).

Can perform automatic variable selection

Weaknesses:

It is a greedy approach to attributes selection.

It is not suited for identifying interaction amongst variables.

Common mistakes:

Cannot be used because of irrelevant attributes.

Cannot work smoothly for missing values.

Overfitting when no pruning.

1. You are given a data set from the automobile industry where the data objects are specific car models and the attributes describe the various features of a car. The attributes are:

Type (Small, Sports, Compact, Midsize, Large)

MPG (miles per gallon)

Drive train type (rear wheel drive, front wheel drive, all wheel drive)

Number of cylinders

Engine size (liters)

Horsepower

RPM (revs per minute at maximum horsepower)

Length (inches)

Wheelbase (inches)

*Class:* Low Price, Average Price, High Price

You are also given training data which has the above information for many car models. The task assigned to you is to build a classification model with the training data so that given a new car model, your model can predict whether it should have a low, average or high price.

Taking into account factors such as interpretability of the model and the presence of domain knowledge, answer the questions below:

* 1. What aspects of this problem would lead you to use decision trees over the kNN classifier?

While many observations about the data set may lead us to choose decision trees over kNN classifier, two of them are of key importance. First, kNN classification assigns the same importance to all attributes, while decision trees can select important attributes. In the data set, some attributes are more important than others. Second, kNN needs a distance function, which is in general non-trivial to define for nominal or ordinal attributes.

* 1. If the data set has additional 5 irrelevant attributes, (e.g. the color of the car), what impact will this have on the relative suitability of naïve Bayes and kNN?

Since irrelevant attributes will not influence the class conditional probabilities, the performance of naïve Bayes wouldn’t be affected. On the other hand, the similarity between different examples will be adversely affected by these irrelevant attributes, thus leading to a fall in the performance of kNN.

* 1. Mention two aspects of this problem that would lead you to use RIPPER over kNN?

Several of these attributes may not be relevant for classification but will adversely affect the similarity computation within kNN. RIPPER will not be affected by these attributes since it performs attribute selection.

Also, the similarity computation in kNN can be adversely affected if different nominal and ordinal attributes have widely varying ranges. On the other hand, RIPPER can simply figure out a threshold for each of the attributes, and thus is not affected by the wide variability in the ranges of some of the attributes.

* 1. What aspects of this problem would lead you to NOT use Naïve Bayes?

Naïve Bayes assumes that the attributes are conditionally independent given the class. Some of the attributes in the data set are dependent: engine size (in liters) correlates with horsepower, or sports cars typically have smaller lengths than large cars.

1. Figures A, B and C below show different scenarios with two target classes (represented as red circles and blue triangles). Please determine which of the following classifiers can work best on each of the figures: (1) Linear Support Vector Machine, (2) K-Nearest Neighbor, and (3) Decision Tree. Give a very brief justification for each case.

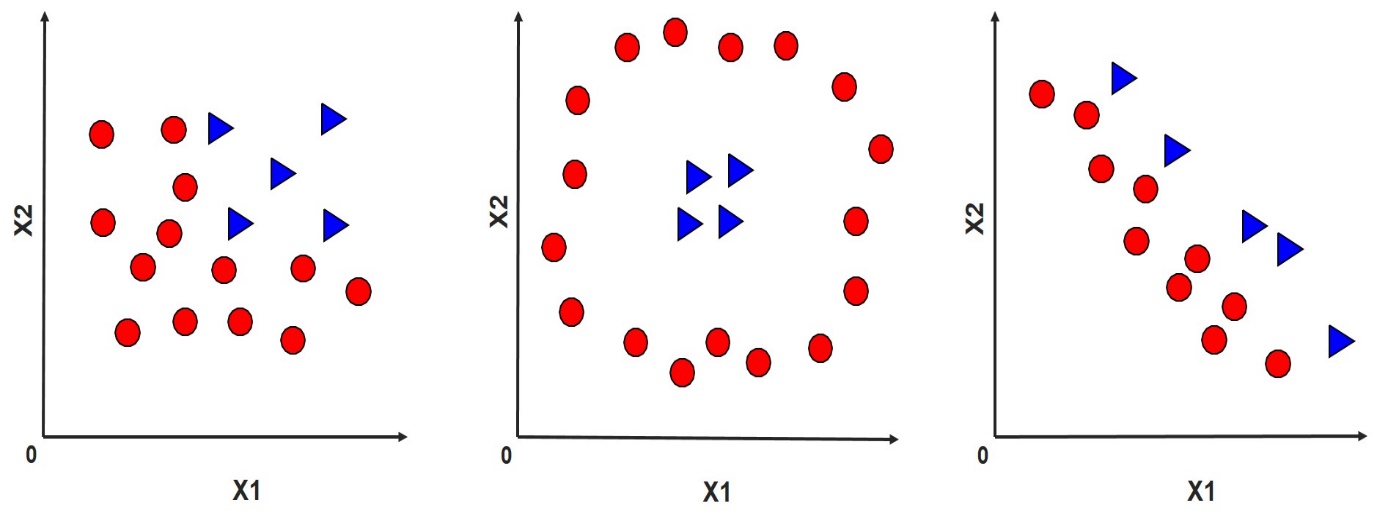


Figure A Figure B Figure C

|  |  |
| --- | --- |
| Figure A | (3) Decision Tree works best because of the rectangular boundary. |
| Figure B | (2) 1-NN works best because, for every point, its nearest neighbor is in the same class with it. Linear SVM will not handle this case well. DT may work but not the best in this case. |
| Figure C | (1) Linear SVM works best because of the linear boundary. Both DT and 1-NN work poorly in this case. |

1. The figure below shows a two-dimensional data set with two target classes (represented as circles and triangles) and the classification boundaries produced by each of the following classifiers: (1) Linear Support Vector Machine, (2) 1- Nearest Neighbor classifier and (3) Decision Tree. Match the classifiers (1), (2) and (3) with the corresponding classification boundaries in Figure 1 (A), (B) and (C) by filling the following table (No justification needed).

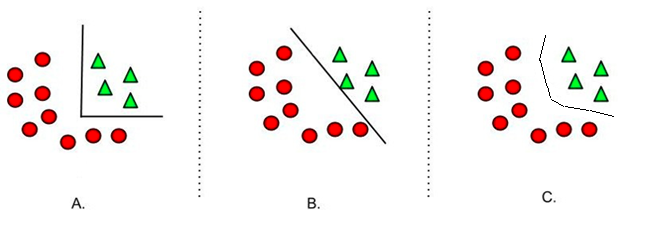


Figure 1: Two-dimensional dataset with classification boundaries from different classifiers

|  |  |
| --- | --- |
|  | Classifier (1,2, or 3) |
| Figure 1(A) | (3) Decision Tree works as it naturally produces a rectangular boundary. |
| Figure 1(B) | (1) Linear SVM, as it naturally produces a linear boundary. |
| Figure 1(C) | (2) K-NN, as it can produce arbitrarily shaped boundaries. |

1. The figure below shows a two-dimensional training set with two target classes (represented as circles and triangles) and four classification boundaries (shown as dotted lines in subfigures A to D) produced by the following classifiers (in no particular order):
2. Linear Support Vector Machine
3. Perceptron
4. 1- Nearest Neighbor classifier
5. Decision Tree.

You need to match these classifiers uniquely to their corresponding classification boundaries. Answer the following questions, providing a one-line justification if requested.

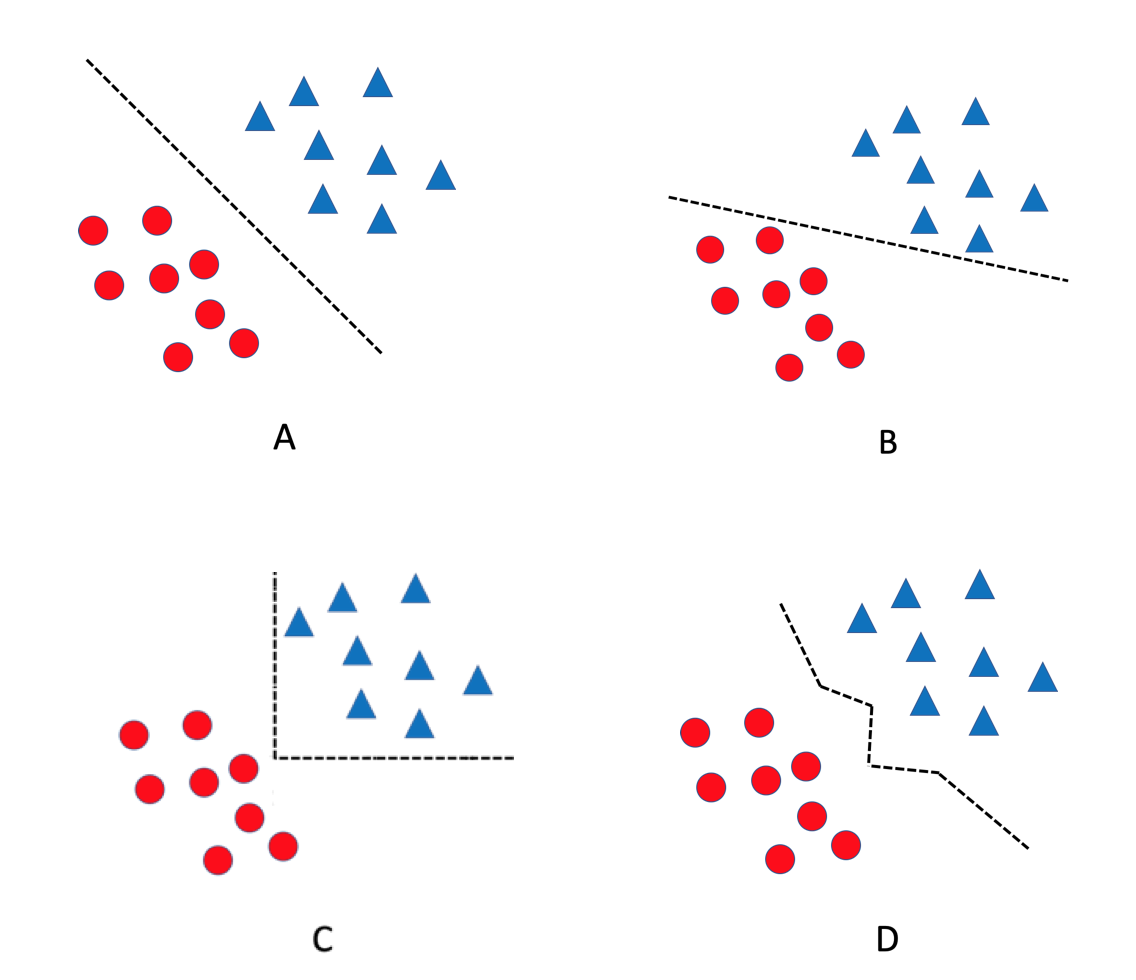


Figure: Two-dimensional dataset with classification boundaries from different classifiers

1. Which classifier matches best with the classification boundary shown in Figure A? Explain.

Answer: SVM. The classification boundary shown in Figure A corresponds to the maximum margin hyper-plane of the data, which can be uniquely determined using the SVM classification technique. While the perceptron learning rule can also, in principle, lead to such a classification boundary, they suffer from local minima and may produce any classification boundary that linearly separates all positive and negative training instances, depending on the initial choice of model parameters.

1. Which classifier matches best with the classification boundary shown in Figure B? Explain.

Answer: Perceptron. We can see that this classification boundary is not the maximum margin hyperplane, and hence cannot be learned by SVM. Decision tree and kNN do not produce oblique but linear decision boundaries.

1. Which classifier matches best with the classification boundary shown in Figure C? Explain.

Answer: Decision trees. The classification boundary consists of rectilinear (axis-parallel) splits, which is characteristic of decision trees.

1. Which classifier matches best with the classification boundary shown in Figure D? No explanation needed.

Answer: kNN.

1. If the training set is such that every combination of attribute values is present in the training data and each combination is either labeled positive or negative, which of the following classification techniques can be used to learn a model with perfect classification (zero errors) on the training set? Briefly explain your answer.
   1. Decision Trees
   2. Logistic Regression
   3. Naïve Bayes
   4. Multi-layer ANN
   5. Perceptron

Answer: Decision Trees and Multi-layer ANN. Both are universal approximators, i.e., we can construct an arbitrarily complex model that can perfectly classify every training instance, provided that the class label for every combination of attribute values is unique.