Introduction to Classification

- Nowadays databases are used for making intelligent decisions.
- Two forms of data analysis namely classification and regression are used for predicting future trends by analyzing existing data.
- Classification models predict **discrete** value or class, while Regression models predict a **continuous** value.
- A **discrete attribute** has a finite or countably infinite set of values, which may or may not be represented as integers while continuous values are real numbers,
- For example, a classification model can be built to predict whether Arsenal will win a football match or not, while regression can be used to predict the number of goals that will be scored by Arsenal in a forthcoming football match.
- Classification is a classical method which is used by machine learning researchers and statisticians for predicting the outcome of **unknown samples**.
- It is used for categorization of objects (or things) into given discrete number of classes. Classification problems can be of two types, either **binary** or **multiclass**.
- In **binary** classification the target attribute can only have two possible values.
- For example, a tumor is either cancerous or not, a team will either win or lose, a sentiment of a sentence is either positive or negative and so on.
- In **multiclass** classification, the target attribute can have more than two values. For example, a tumor can be of type 1, type 2 or type 3 cancer; the sentiment of a sentence can be happy, sad, angry or of love; news stories can be classified as weather, finance, entertainment or sports news
- Some examples of business situations where the classification technique is applied are:
    - To analyze the credit history of bank customers to identify if it would be risky or safe to grant them loans.
    - To analyze the purchase history of a shopping mall's customers to predict whether they will buy a certain product or not.
- In first example, the system will predict a **discrete value** representing either **risky** or **safe**, while in second example, the system will predict **yes** or **no**.
- Some more examples to distinguish the concept of regression from classification are:
    - To predict how much a given customer will spend during a sale.
    - To predict the salary-package of a student that he/she may get during his/her placement.
- In these two examples, there is a prediction of continuous numeric value. Therefore, both are regression problems.

Types of Classification

Classification is defined as two types. These are:

- Posteriori classification
- Priori classification

Posteriori classification

- The word '**Posteriori'** means something derived by reasoning from the observed facts.
- It is a supervised machine learning approach, where the target classes are already known, i.e., training data is already labeled with actual answers.

Priori classification

- The word '**Priori**' means something derived by reasoning from self-evident propositions.
- It is an unsupervised machine learning approach, where the target classes are not given.
- The question is '**Is it possible to make predictions**, if labeled data is not available?'
- The answer is yes. If data is not labeled, then we can use **clustering** (unsupervised technique) to divide unlabeled data into clusters.
- Then these clusters can be assigned some names or labels and can be further used to apply classification to make predictions based on this dataset.
- Thus, although data is not labeled, we can still make predictions based on data by first applying clustering followed by classification.
- This approach is known as Priori classification.

Input and Output Attributes

- Data contains two types of attributes, namely, input attributes and output attributes.
- The class attribute that represents the output of all other attributes is known as an **output** attribute or **dependent** attribute, while all other attributes are known as **input** attributes or **independent** attributes.

The attributes can be of different types.

- The attributes having numbers are called **numerical attributes** while attributes whose domain is not numerical are known as **nominal** or **categorical attributes**.

Class

Input Attributes				Output Attribute
Sepal Length	Sepal Width	Petal Length	Petal Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa

Numerical Nominal

Figure 5.1 Input and output attributes

In figure 5.1, input attributes are of numerical type while species, i.e., the output attribute is of nominal type.

Working of Classification

Classification is a **two-step process**:

- The **first step** is training the model and the **second step** is testing the model for accuracy.
- In the **first step**, a **classifier** is built based on the training data obtained by analyzing database tuples and their associated class labels.
- By analyzing training data, the system learns and creates some rules for prediction.
- In the **second step**, these prediction rules are tested on some unknown instances, i.e., test data.

- In this step, rules are used to make the predictions about the output attribute or class.
- In this step, the predictive accuracy of the classifier is calculated.
- The system performs in an iterative manner to improve its accuracy, i.e., if accuracy is not good on test data, the system will reframe its prediction rules until it gets optimized accuracy on test data.
- The test data is randomly selected from the full dataset.
- The tuples of the test data are independent of the training tuples.
- This means that the system has not been exposed to testing data during the training phase.
- The accuracy of a classifier on a given test data is defined as the percentage of test data tuples that are correctly classified by the classifier.
- The associated class label of each test tuple is compared with the class prediction made by the classifier for that particular tuple.
- If the accuracy of the classifier is satisfactory then it can be used to classify future data tuples with unknown class labels.

For example, by analyzing the data of previous loan applications as shown, the classification rules obtained can be used to approve or reject the new or future loan applicants as shown.

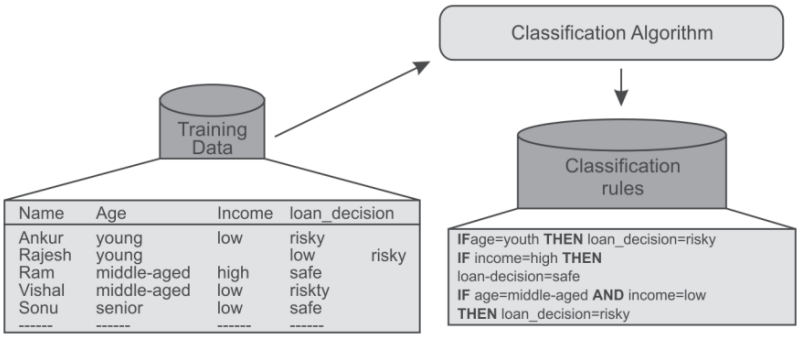


Figure 5.3 Building a classifier to approve or reject loan applications

Step 2

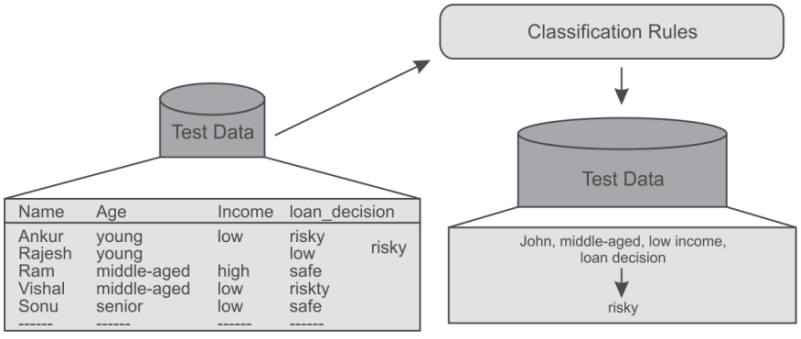
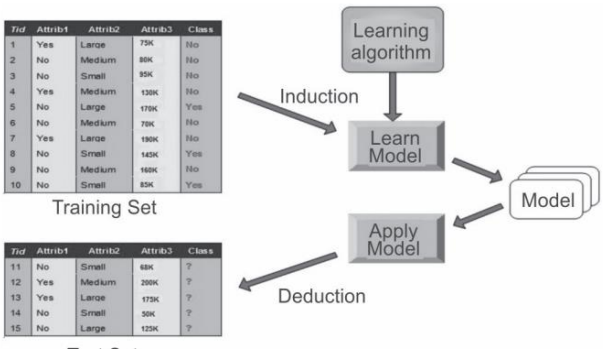


Figure 5.4 Predicting the type of customer based on trained classifier

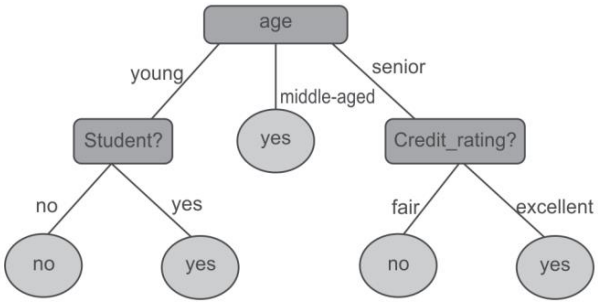
The same process of training and testing the classifier has also been illustrated:



Training and testing of the classifier

Introduction to the Decision Tree Classifier

- In the decision tree classifier, predictions are made by using multiple 'if…then…' conditions which are similar to the control statements in different programming languages.
- The decision tree structure consists of a **root node**, **branches** and **leaf nodes**.
- Each **internal node** represents a **condition** on some input attribute, each **branch** specifies the **outcome of the condition** and each **leaf node** holds a **class label**.
- The root node is the topmost node in the tree.
- The decision tree shown represents a classifier tasked for predicting whether a customer will buy a laptop or not. Here, each internal node denotes a **condition** on the input attributes and each leaf node denotes the predicted outcome (class).
- By traversing the decision tree, one can analyze that if a customer is middle aged then he will probably buy a laptop, if a customer is young and a student then he will probably not buy a laptop.
- If a customer is a senior citizen and has an excellent credit rating then he can probably buy a laptop.
- The system makes these predictions with a certain level of probability.



- Decision trees can easily be converted to classification rules in the form of if-then statements.
- Here, each node of the decision tree denotes a choice between numbers of alternatives and the choices are binary.
- Each leaf node specifies a decision or prediction.
- The training process that produces this tree is known as induction.

Building decision tree

- A decision tree algorithm known as ID3 (Iterative Dichotomiser) and C4.5 (a successor of ID3) are commonly used to implement classification.
- These algorithms are based on the concept of Information Gain and Gini Index.

Concept of Information Theory

- Decision tree algorithm works on the basis of information theory.
- It has been observed that **information** is directly related with **uncertainty**.
- If there is **uncertainty** then there is **information** and if there is **no uncertainty** then there is no **information**.
- For example, if a coin is **biased** having a head on both sides, then the result of tossing it does **not give any information** but if a coin is *unbiased* having a head and a tail then the result of the toss provides *some information.*
- Let us consider another example, if in Maseno university, there is holiday on Sunday then a notice regarding the same will not carry any information (because it is certain) but if some particular Sunday becomes a working day then it will be information and henceforth becomes a news.
- From these examples we can observe that **information** is related to *the probability of occurrence* of an event.
- Another important question to consider is, whether the probability of occurrence of an event is more. Then, the ***information gain*** will be **more frequent** or **less frequent**?
- It is certain from above examples that 'more certain' events such as Sunday being a holiday carry very little information. But if Sunday is working, then even though the probability of these events is lesser than the previous event, it will carry more information.
- Hence, **less probability** means **more information**.

Defining Information in Terms of Probability

- Information theory was developed by Claude Shannon.
- Information theory defines **entropy** which is average amount of information given by a source of data. Entropy is measured as follows.

  entropy $(p1, p2, \ldots, pn) = -p1\log(p1) - p2\log(p2) - \ldots\ldots - pn\log(pn)$
- Therefore, the total information for an event is calculated by the following equation:

  $\mathbf{I} = \sum_i (-\mathbf{P}_i \log \mathbf{P}_i)$
- In this, information is defined as $-p_i \log p_i$ where p_i is the probability of some event.
- Since, probability p_i is always less than 1, log pi is always negative; thus, negating log p_i we get the overall information gain ($-p_i \log p_i$) as positive.
- It is important to remember that the logarithm of any number greater than 1 is always positive and the logarithm of any number smaller than 1 is always negative.
- Logarithm of 1 is always zero, no matter what the base of logarithm is.

In case of log with base 2, following are some examples.

$$\log_2(2) = 1$$

$\log_2(2^n) = n$

$\log_2(1/2) = -1$

$\log_2(1/2^n) = -n$

- Let us calculate the information for the event of throwing a coin.
- It has two possible values, i.e., head (p_1) or tail (p_2).
- In case of unbiased coin, the probability of head and tail is 0.5 respectively.
- Thus, the information is

$I = -0.5 \log(0.5) - 0.5 \log(0.5)$

$= - (0.5) * (-1) - (0.5) * (-1)$ [As, $\log_2 (0.5) = -1$]

$= 0.5 + 0.5 = 1$

- The result is 1.0 (using log base 2) and it is the maximum information that we can have for an event with two possible outcomes.
- This is also known as entropy.
- But if the coin is biased and has heads on both the sides, then probability for head is 1 while the probability of tails will be 0.
- Thus, total information in tossing this coin will be as follows.
  - $I = -1 \log(1) - 0 \log(0) = 0$ [As, $\log_2(1) = 0$]
- You can clearly observe that tossing of biased coin carries no information while tossing of unbiased coin carries information of 1.

Note

- Information plays a key role in selecting the root node or attribute for building a decision tree.
- In other words, selection of a split attribute plays an important role.
- Split attribute is an attribute that reduces the uncertainty by largest amount, and is always accredited as a root node.
- So, the attribute must distribute the objects such that each attribute value results in objects that have as little uncertainty as possible.
- Ideally, each attribute value should provide us with objects that belong to only one class and therefore have zero information.

Information Gain

- Information gain specifies the amount of information that is gained by knowing the value of the attribute.
- It measures the 'goodness' of an input attribute for predicting the target attribute.
- The attribute with the highest information gain is selected as the next split attribute.
- Mathematically, it is defined as the entropy of the distribution before the split minus the entropy of the distribution after split.
- **Information gain** = (Entropy of distribution before the split) – (Entropy of distribution after the split)
- The **largest information gain** is equivalent to the **smallest entropy** or **minimum information**.

- It means that if the result of an event is **certain**, i.e., the probability of an event is **1** then information provided by it is **zero** while the **information gain** will be the **largest**, thus it should be selected as a **split attribute**.
- Assume that there are two classes, P and N, and let the set of training data S (with a total number of records *s*) contain *p* records of class P and *n* records of class N. The amount of information is defined as

  $I = -(p/s)\log(p/s) - (n/s)\log(n/s)$

- Thus after computing the information gain for every attribute, the attribute with the highest information gain is selected as split attribute.

Building a Decision Tree for the Example Dataset

Let us build decision tree for the following dataset:

INSTANCE NUMBER	X	Y	Z	CLASS
1	1	1	1	A
2	1	1	0	A
3	0	0	1	B
4	1	0	0	B

X	Y	Z	CLASS
1 =3	1=2	1=2	A=2
0=1	0=2	0=2	B=2

Dataset for class C prediction based on given attribute condition

- The given dataset has three input attributes X, Y, Z and one output attribute Class.
- The instance number has been given to show that the dataset contains four records (basically for convenience while making references).
- The output attribute or class can be either A or B.
- There are two instances for each class so the frequencies of these two classes are given as follows:
  - A = 2 (Instances 1, 2)
  - B = 2(Instances 3, 4)

The amount of information contained in the whole dataset is calculated as follows:

I = **- probability for Class A * log (probability for class A)**

- probability for class B * log (probability for class N)

Here, *probability for class A = (Number of instances for class A/Total number of instances) = 2/4*

And *probability for class B = (Number of instances for class B/Total number of instances) = 2/4*

Therefore, I = (-2/4) log (2/4) – (2/4) log (2/4) = 1

Let us consider each attribute one by one as a split attribute and calculate the information for attribute.

Attribute 'X'

- As given in the dataset, there are two possible values of X, i.e., 1 or 0.
- Let us analyze each case one by one.
    - For X= 1, there are 3 instances namely instance 1, 2 and 4. *The first two instances are labeled as class A and the third instance, i.e, record 4 is labeled as class B.*
    - For X = 0, there is only 1 instance, i.e, *instance number 3 which is labeled as class B.*

Given the above values, let us compute the information given by this attribute. We divide the dataset into two subsets according to X either being 1 or 0. Computing information for each case,

I (for X = 1) = I(X1) = - (2/3) log(2/3) - (1/3) log(1/3) = 0.92333

I (for X = 0) = I(X0) = - (0/1) log(0/1) - (1/1) log(1/1) = 0

Total information for above two sub-trees *= probability for X having value 1 * I(X1) + probability for X having value 0 * I(X0)*

Here, probability for X having value 1 = *(Number of instances for X having value 1/Total number of instances) = 3/4*

And probability for X having value 0 = *(Number of instances for X having value 0/Total number of instances) = 1/4*

Therefore, total information for the two sub-trees = (3/4) I(X1) + (1/4) I(X0)

= 0.6925 + 0

= 0.6925

Attribute 'Y'

- There are two possible values of Y attribute, i.e., 1 or 0.
- Let us analyze each case one by one.
    - There are 2 instances where Y has value 1. In both cases when Y=1 the *record belongs to class A* and,
    - in the 2 instances when Y = 0 both *records belong to class B.*

Given the above values, let us compute the information provided by Y attribute.

We divide the dataset into two subsets according to Y either being 1 and 0. Computing **information** for each case,

I (For Y = 1) = I (Y1) = - (2/2) log(2/2) - (0/2) log(0/2) = 0

I (For Y = 0) = I (Y0) = - (0/2) log(0/2) - (2/2) log(2/2) = 0

Total information for the two sub-trees *= probability for Y having value 1* I(Y1) + probability for Y having value 0 * I(Y0)*

Here, *probability for Y in 1 = (Number of instances for Y having 1/Total number of instances) = 2/4*

And probability for Y in 0 = *(Number of instances for Y having 0/Total number of instances) = 2/4*

Therefore, the total information for the two sub-trees

I= (2/4) I(Y1) + (2/4) I(Y0)

= 0 + 0

= 0

Attribute 'Z'

- There are two possible values of Z attribute, i.e., 1 or 0.
- Let us analyze each case one by one.
    - There are 2 instances where Z has *value 1 and 2 instances where Z has value 0.*
    - *In both cases, there exists a record belonging to class A and class B with Z is either 0 or 1.*

Given the above values, let us compute the information provided by the Z attribute.

We divide the dataset into two subsets according to Z either being 1 or 0. Computing information for each case,

I (For Z = 1) = I (Z1) = - (1/2) log (1/2) - (1/2) log (1/2) = 1.0

I (For Z = 0) = I (Z0) = - (1/2) log (1/2) - (1/2) log (1/2) = 1.0

Total information for the two sub-trees *= probability for Z having value 1 * I(Z1) + probability for Z having value 0 * I(Z0)*

Here, *probability for Z having value 1 = (Number of instances for Z having value 1/Total number of instances) = 2/4*

And *probability for Z having value 0 = (Number of instances for Z having value 0/Total number of instances) = 2/4*

Therefore, total information for two sub-trees

= (2/4) I(Z1) + (2/4) I(Z0)

= 0.5 + 0.5

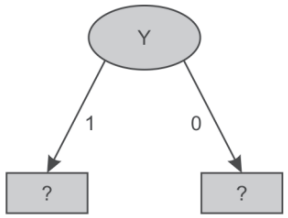
= 1.0

The Information gain can now be computed:

Potential Split attribute	Information before split	Information after split	Information gain
X	1	0.6925	0.3075

Y	1	0	1.0
Z	1	1	0

Hence, the largest information gain is provided by the attribute 'Y' thus it is used for the split



Data splitting based on Y attribute

For Y, as there are two possible values, i.e., 1 and 0, therefore the dataset will be split into two subsets based on distinct values of the Y attribute as shown:

Dataset for Y = '1'

Instance	X	Z	Class
1	1	0	A
2	1	1	A

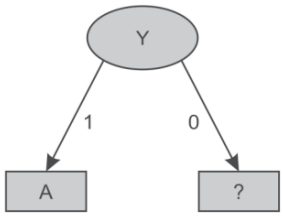
There are 2 samples and the frequency of each class is as follows.

- A = 2 (Instances 1, 2)
- B = 0 Instances

Information of the whole dataset on the basis of class is given by

$$I = (-2/2) \log (2/2) - (0/2) \log(0/2) = 0$$

As it represents the same **class 'A'** for all recorded combinations of X and Z, therefore, it represents *class 'A'* as shown:

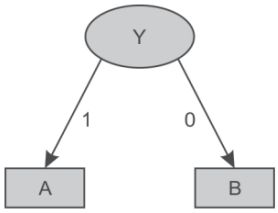


- Decision tree after splitting of attribute Y having value '1'

Dataset for Y = '0'

Instance	X	Z	Class
3	1	0	B
4	0	1	B

For **Y having value 0,** it represents the same *class 'B'* for all the records. Thus, the decision tree will look like as shown after analysis of Y dataset.



- Decision tree after splitting of attribute Y value '0'

Let us consider another example and build a decision tree for the dataset given.

Instance Number	Outlook	Temperature	Humidity	Windy	Play
1	sunny	hot	high	false	No
2	sunny	hot	high	true	No
3	overcast	hot	high	false	Yes
4	rainy	mild	high	false	Yes
5	rainy	cool	normal	false	Yes
6	rainy	cool	normal	true	No
7	overcast	cool	normal	true	Yes
8	sunny	mild	high	false	No
9	sunny	cool	normal	false	Yes
10	rainy	mild	normal	false	Yes
11	sunny	mild	normal	true	Yes
12	overcast	mild	high	true	Yes
13	overcast	hot	normal	false	Yes
14	rainy	mild	high	true	No

Attribute values and counts

Outlook	Temp.	Humidity	Windy	Play
sunny = 5	hot = 4	high = 7	true = 6	yes = 9
overcast = 4	mild = 6	normal = 7	false = 8	no = 5
rainy = 5	cool = 4			

- It has 4 input attributes *outlook*, *temperature*, *humidity* and *windy*.
- As before we have added instance number for explanation purposes.
- Here, 'play' is the output attribute and these 14 records contain the information about weather conditions based on which it was decided if a play took place or not.
- In the dataset, there are 14 samples and two classes for target attribute 'Play', i.e., Yes or No. The frequencies of these two classes are given as follows:
    - Yes = 9 (Instance number 3,4,5,7,9,10,11,12,13 and 14)
    - No = 5 (Instance number 1,2,6,8 and 15)