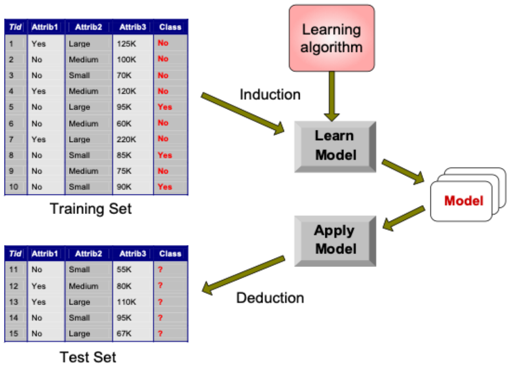
Classification and Decision Trees Activity

Activity 1 : Supervised Learning

Given a set of example data, which we will call *training data*, the goal is to learn a function/model that will compute a target value for new data. This is known as **supervised** learning. An example of supervised learning is predicting the sales price of a home. Example data for this task is shown in the table below.

SQFT	# of bedrooms	zip code	Sales price
1000	2	22999	160,000
2000	4	20111	450,000
2200	3	90210	850,000



Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

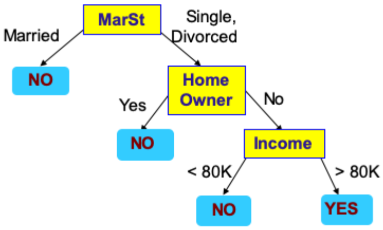
Introduction to Data Mining. Pang-Ning Tan et. al. 2019.

When the predicted value is a real number (sales price), we call this *regression*. When the predicted value is a label, this is called *classification*, for example, identifying an email as **spam** or **not spam**. When only two labels are possible (as is the case with spam), this is called *binary classification*. A general approach to classification to shown on the left.

1. List 3 possible uses of **regression**.

2. List 3 possible uses of **classification**.

Activity 2 : Tree Structures for Classification



A tree structure can be created to perform classification. For example, the tree on the left is used to predict whether someone who borrows money from a bank will default on the loan.

1. Use the tree to classify a newly arrived example with the following features:

Rowid	Homeowner	Marital Status	Income	Loan Will Default?
1	no	married	80K	??

   Trace the pathway through the tree. What label (yes or no) does the tree assign?

Decision Tree Construction

The following is a subset of the Titanic Survivors dataset available from Kaggle `https://www.kaggle.com/c/titanic/`. Draw a decision tree that correctly classifies each item.

Pclass	Sex	Age	Survived
1	female	30	Yes
3	male	25	No
3	male	33	No
2	male	34	No
2	female	45	Yes
2	male	44	No
1	male	27	Yes
2	female	31	Yes
2	male	34	No
3	male	23	No

Now use your tree to classify the items in this test set:

Pclass	Sex	Age	Survived
1	female	17	
3	female	41	
3	male	17	
3	male	1	

Tree Reflection

In this case, it was straightforward to develop a decision tree that could perfectly classify the training set by “eyeballing” the data. Consider the case where we have thousands of elements in our training set, each with dozens of attributes.

- How could you automate the process of tree construction?

- Given that there are potentially many possible trees that fit the training data, how might you go about deciding between them? What constitutes a “good” tree?