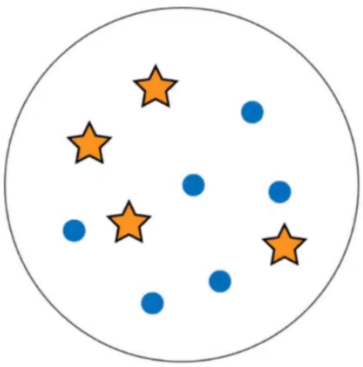
Introduction to Classification

Introduction to Business Analytics

Probability 101

★ = Customer made a purchase

● = Customer did not make a purchase



$$P(★) = \frac{\#\ Customers\ ★}{Total\ \#\ Customers} = 0.4$$

$$P(●) = \frac{\#\ Customers\ ●}{Total\ \#\ Customers} = 0.6$$

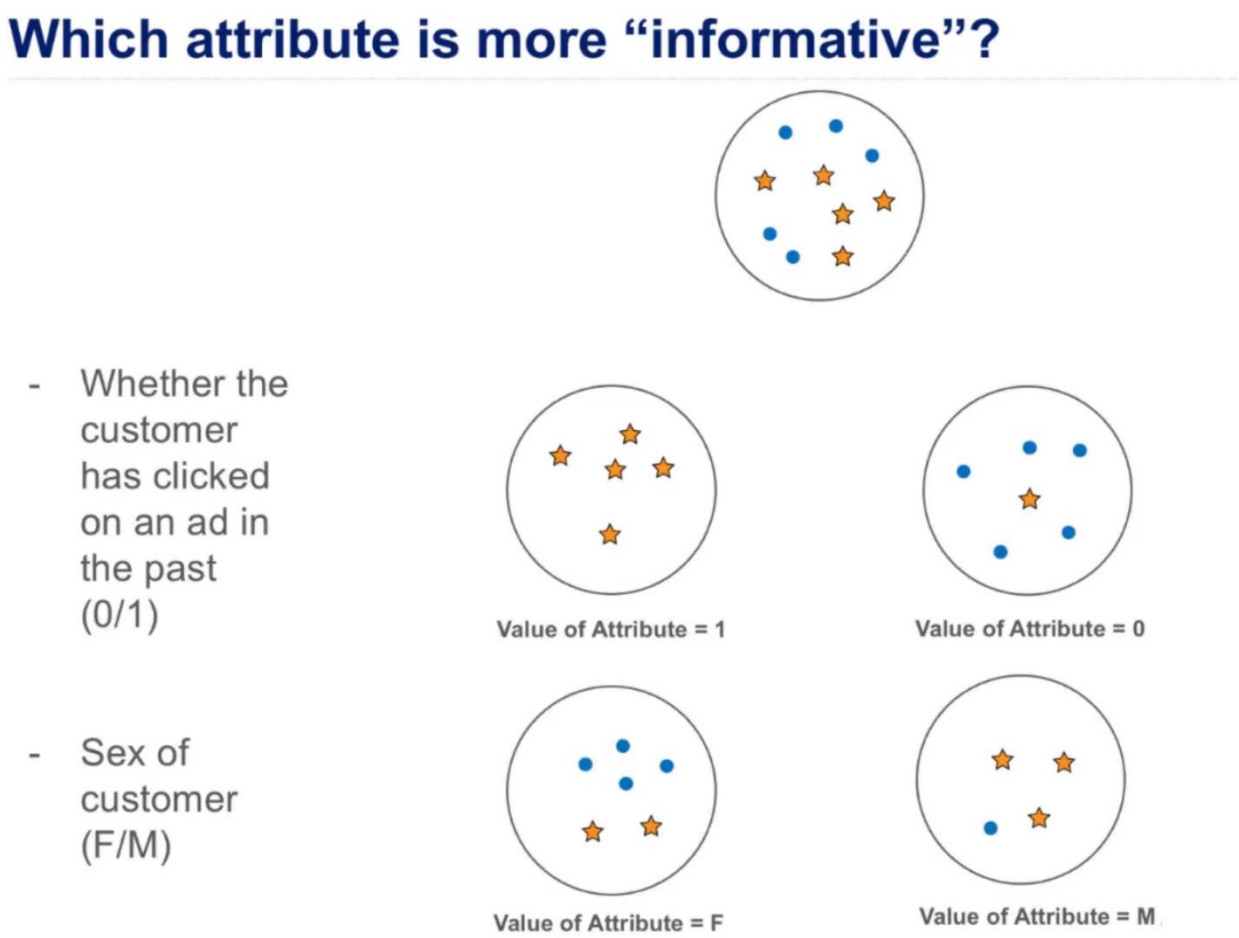
Data Set

Target Variable

★ Conversion = 1
● Conversion = 0

Attributes / Features

Conversion	Sex	Clicked on Ad
1	F	1
0	F	0
0	M	0
1	M	1
1	F	0
1	M	1
0	F	0
1	M	1
0	F	0
0	F	0



- All people who clicked on the add purchased the product
- Out of all the people who didn't click the ad, only one person purchased the product
- Out of all the female samples, only two of them purchased the product
- Out of all the male samples, three of them purchases the product

The first attribute, which tells us whether the customers has clicked on an ad in the past, tells us a better sense of whether conversion happened or not.

Supervised Segmentation

- How can we **segment** the population into groups that differ from each other *with respect to some* ***quantity of interest***?
- Informative attributes
  - Find **knowable** attributes that **correlate** with the **target of interest**
    - Increase accuracy
    - Alleviate computational problems
    - e.g. *tree induction*
- How can we judge whether a variable contains important information about the target variable?
  - How much?

The goal of the segmentation problem is **to represent images in a simplified and meaningful manner by partitioning them into multiple groups.**

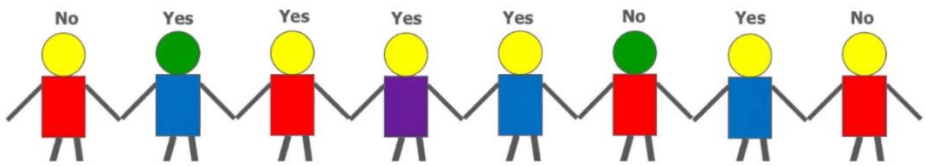
Practical applications of image segmentation include medical imaging, video surveillance, as well as means of pre-processing for a magnitude of detection and recognition tasks.

Supervised segmentation algorithms use a priori knowledge involving the ground truth of a training set of images, while **unsupervised algorithms** are trained online during segmentation.

The more information the attribute provides, the more uncertainty is being reduced. We need to know how we can quantify how much information does each attribute contains.

Selecting Informative Attributes

Objective: Based on customer attributes, partition the customers into **subgroups** that are **less impure** – with respect to the class (i.e., such that in each group as many instances as possible belong to the same class)



There are two types of hat: yellow and green.

There are two types of values: red and blue.

There are two target variables: Yes and No.

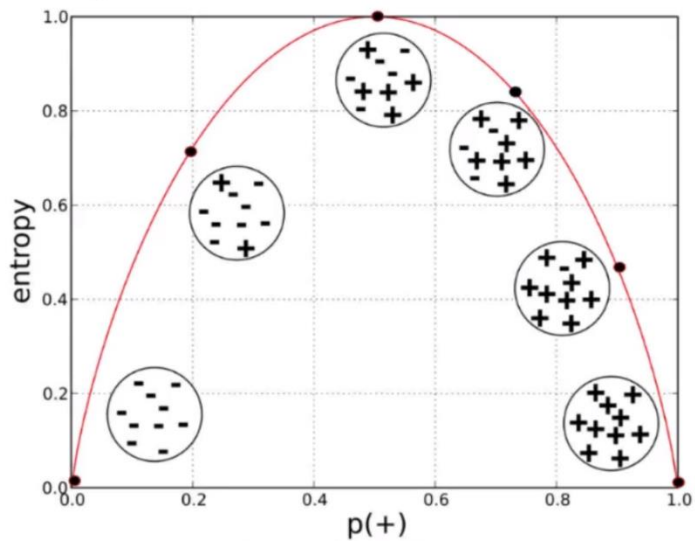
The important question is, "Which of these attributes could be best to segment this people into groups?". Technically, there are several complications as there could be numeric attributes where it is no longer a binary variable.

Selecting Informative Attributes

- One of the most common splitting criterion is called **information gain** (IG)

Information gain is the **reduction in entropy or surprise by transforming a dataset** and is often used in training decision trees. Information gain is calculated by comparing the entropy of the dataset before and after a transformation. Information gain evaluates and quantifies **how well each attribute is split into segments with respect to the target variable**.

- It is based on a **purity measure** called **entropy**
  - $entropy = -\mathrm{p}_1 \log_2(p_1) - p_2 \log_2(p_2) - ..$
  - Measures the general disorder of a set



0 in entropy scale indicates this set is completely **pure** and **homogeneous**, meaning all the customers belonging in the same class. In the case, where entropy is 0, we can be certain about target variable.

1 is the maximum value that entropy can take. And this value indicates that segmentation is completely **impure** or **heterogeneous**, meaning 50 percent of the customers belong in the positive class and other 50 percent belong in the negative class.

Values between 0 and 1 indicates the possible degrees of purity between segmentations. To estimate the entropy of the set, we use the formula above.

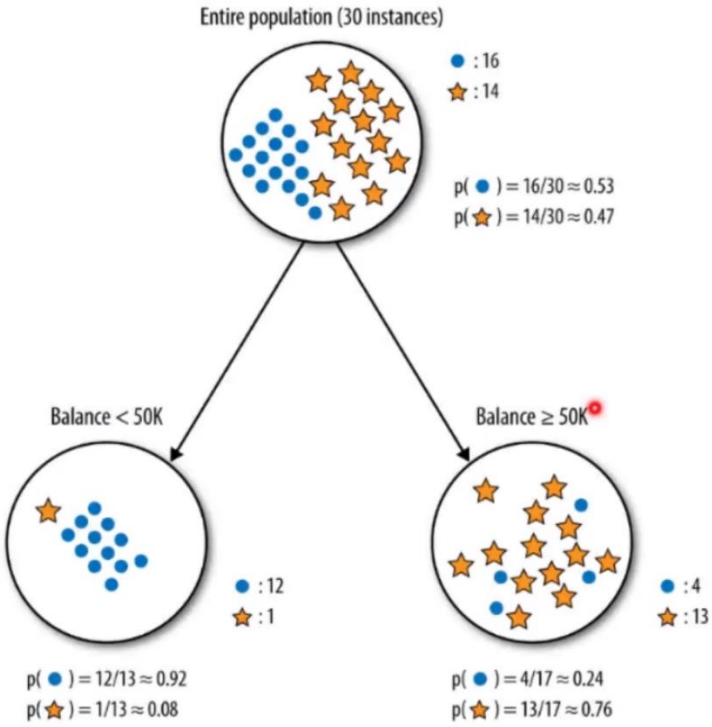
Assuming we are dealing with binary variables, p1 indicates the possibility of making purchase and p2 indicates the possibility of not making the purchase. The equation above is equivalent to the following:

$$-[\mathrm{p}_1 \log_2(p_1) + p_2 \log_2(p_2) + ..]$$

Information Gain

- Information gain measures the ***change* in entropy** due to any amount of new information being added

Even if attribute is continuous, we can make it categorical by choosing a specific threshold.

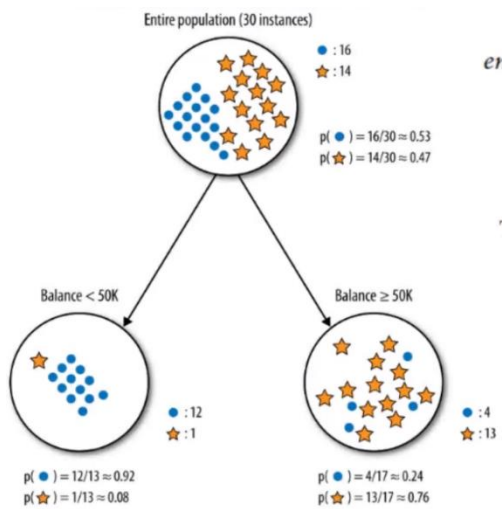


$$IG(parent, children) = entropy(parent) - [p(c_1) \times entropy(c_1) + p(c_2) \times entropy(c_2) + \cdots]$$

Parent is the initial set of population that we began with. We take the entropy of the parent and subtract the entropy of the children, all of which enclosed in a square bracket. C1 denotes the left child and c2 denotes the right child.

If I use, for example, the attribute 'balance' to segment the customers into the different subgroups, how much information do I gain with respect to the target variable?

P(c1) is 13/30 and P(c2) is 17/30. To sum up, we can quantify the Information Gain of the specific attribute by subtracting from the entropy of the parent the weighted average of entropy of the children.



$$\begin{aligned} entropy(parent) &= -[p(\bullet) \times \log_2 p(\bullet) + p(\star) \times \log_2 p(\star)] \\ &\approx -[0.53 \times -0.9 + 0.47 \times -1.1] \\ &\approx 0.99 \quad \text{(very impure)} \end{aligned}$$

The entropy of the *left* child is:

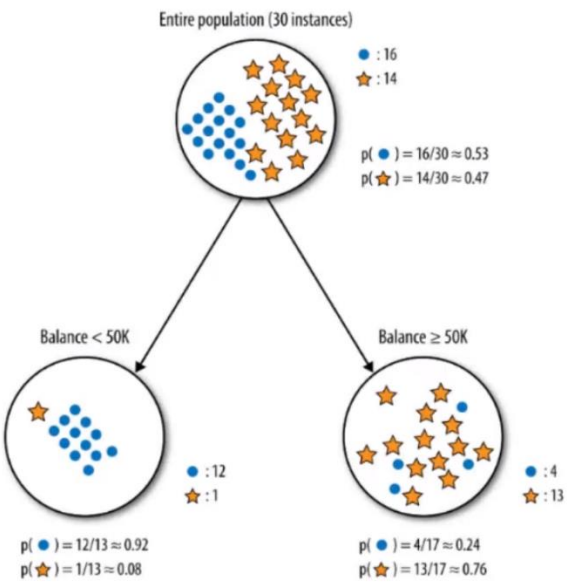
$$\begin{aligned} entropy(Balance < 50K) &= -[p(\bullet) \times \log_2 p(\bullet) + p(\star) \times \log_2 p(\star)] \\ &\approx -[0.92 \times (-0.12) + 0.08 \times (-3.7)] \\ &\approx 0.39 \end{aligned}$$

The entropy of the *right* child is:

$$\begin{aligned} entropy(Balance \geq 50K) &= -[p(\bullet) \times \log_2 p(\bullet) + p(\star) \times \log_2 p(\star)] \\ &\approx -[0.24 \times (-2.1) + 0.76 \times (-0.39)] \\ &\approx 0.79 \end{aligned}$$

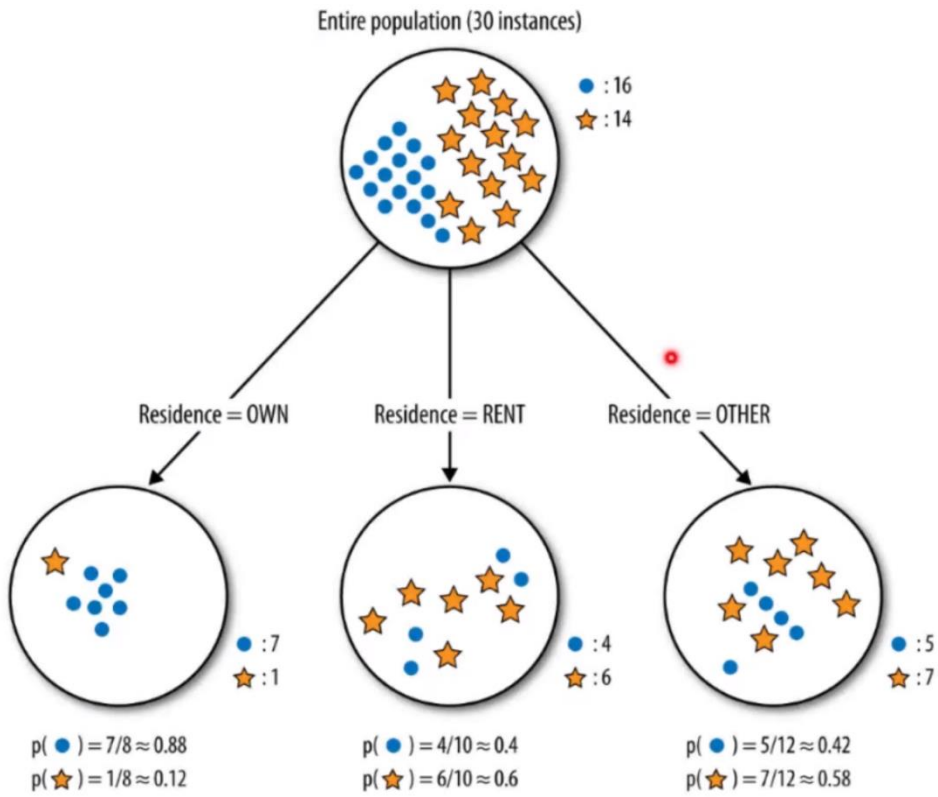
Child group is purer than parent group as indicated by its entropy value.

Information Gain



$$\begin{aligned} IG &= entropy(parent) - [p(\text{Balance} < 50\text{K}) \times entropy(\text{Balance} < 50\text{K}) \\ &\quad + p(\text{Balance} \geq 50\text{K}) \times entropy(\text{Balance} \geq 50\text{K})] \\ &\approx 0.99 - [0.43 \times 0.39 + 0.57 \times 0.79] \\ &\approx 0.37 \end{aligned}$$

IG value itself doesn't tell a lot, but comparing two values would be great in terms of **which attribute is more informative**. Ideally, we want to use the threshold that maximizes our IG.



Let's use another set of criteria, residence status, to split parent group into three child groups.

$$
\begin{aligned}
entropy(parent) &\approx 0.99 \\
entropy(\text{Residence=OWN}) &\approx 0.54 \\
entropy(\text{Residence=RENT}) &\approx 0.97 \\
entropy(\text{Residence=OTHER}) &\approx 0.98 \\
IG &\approx 0.13
\end{aligned}
$$

IG is 0.13, which is lower than that of the previous one. Thus, it can't be more useful. Balance is more informative attribute than the residence status.

Information gain is a measure of how much an attribute improves or decreases the entropy over the whole segmentation it creates.

Attribute Selection

Reasons for selecting only a subset of attributes:

- Better **insights** and business understanding
- Better **explanations** and more tractable models
- Reduced **cost**
- **Faster** predictions
- **Better** predictions!
  - Over-fitting (*to be continued*..)

Example: Attribution Selection with Information Gain

- This dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family.
- Each species is identified as definitely *edible*, definitely *poisonous*, or of *unknown edibility and not recommended*.
  - This latter class was combined with the poisonous one.
- The Guide clearly states that there is **no simple rule for determining the edibility of a mushroom**; no rule like "leaflets three, let it be" for Poisonous Oak and Ivy.

1

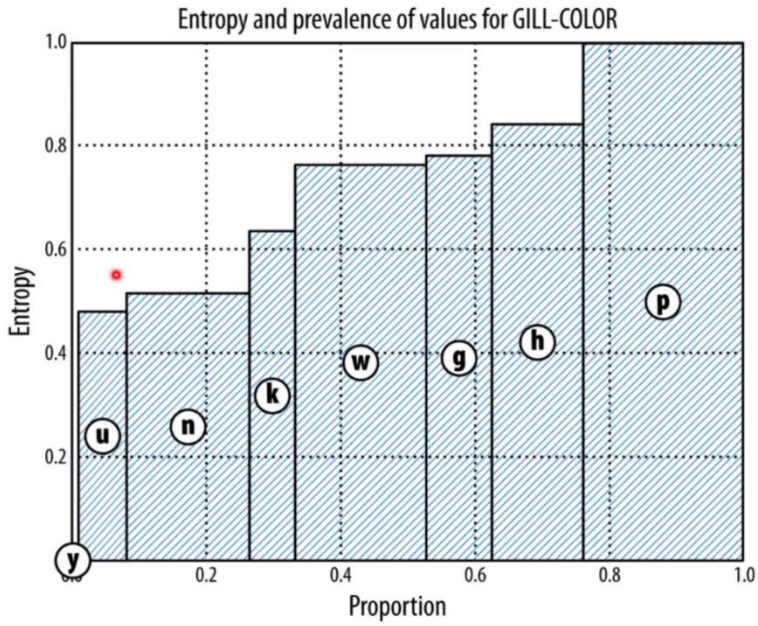
Example: Attribution Selection with Information Gain

Attribute name	Possible values
CAP-SHAPE	bell, conical, convex, flat, knobbed, sunken
CAP-SURFACE	fibrous, grooves, scaly, smooth
CAP-COLOR	brown, buff, cinnamon, gray, green, pink, purple, red, white, yellow
BRUISES?	yes, no
ODOR	almond, anise, creosote, fishy, foul, musty, none, pungent, spicy
GILL-ATTACHMENT	attached, descending, free, notched
GILL-SPACING	close, crowded, distant
GILL-SIZE	broad, narrow
GILL-COLOR	black, brown, buff, chocolate, gray, green, orange, pink, purple, red, white, yellow
STALK-SHAPE	enlarging, tapering
STALK-ROOT	bulbous, club, cup, equal, rhizomorphs, rooted, missing
STALK-SURFACE-ABOVE-RING	fibrous, scaly, silky, smooth
STALK-SURFACE-BELOW-RING	fibrous, scaly, silky, smooth

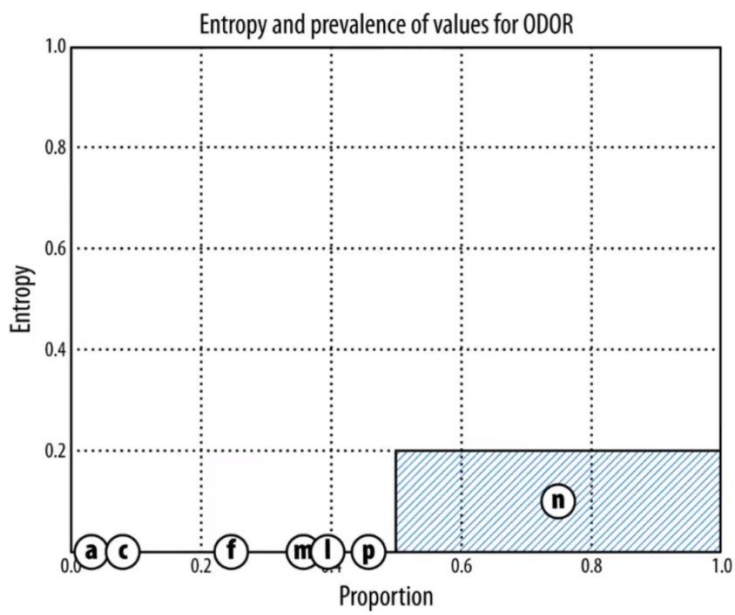
There are lots of attributes we can use to segment mushroom into different subgroups. If we were to choose single attribute to determine the type of mushroom with high level of accuracy, what would we choose?

It turns out that information gain is a great way to predict whether mushroom is edible or not.

Example: Attribution Selection with Information Gain



White areas indicate Information Gain. With the attribute, GILL-COLOR, we have gained some IG but it is not substantial.



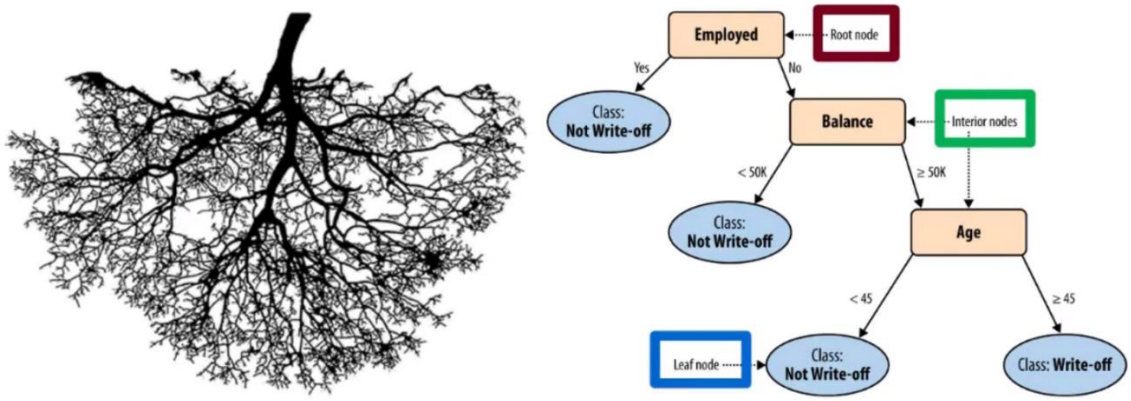
Entropy of the subgroups are so much lower with the attribute, ODOR, indicating the higher information gain and thus tells ODOR is more informative attribute than GILL-COLOR in terms of predicting whether the mushroom is poisonous or not.

Multivariate Supervised Segmentation

- If we select the ***single* variable** that gives the most information gain, we create a very *simple* segmentation.
- If we **select multiple attributes** each giving some information gain, how do we put them together?

How do we combine these features to give a multivariate supervised segmentation?

Tree-Structured Models



Tree is upside-down with root at the top. It is made up of root node, interior node, and terminal (leaf) node.

Tree creates a segmentation of the data. Every data point in our dataset, we will correspond to one and only one path in the tree and thereby one and only one leaf node. Each leaf corresponds to a segment. And path along the yellow one gives the characteristics of the segment.

For example, people who are in the class, 'Not Write-off' belongs to unemployed, balance of more than 50k, and less than 45 years old. This is considered a supervised segmentation because each leaf contains the prediction for the target variable.

For the initial population, attribute, 'Employed' is the most informative attribute as it has the highest IG value. Thus, we used this attribute to create our first set of segmentation. Remember that two set of attributes are created because 'Employed' is a binary variable.

'Balance' attribute is still quite impure and heterogeneous, so we want to create another segmentation. We have to re-examine the Information Gain for each attribute. Even though 'Balance' is a numeric variable, we can choose the threshold that will give the highest IG value.

Yet, we select another attribute, 'Age' to create more pure and homogenous segmentation by selecting the next highest IG attribute. 'Age' is again continuous attribute, and we want to set a threshold that will provide the highest IG segmentation, after which the attribute is pure enough that no further segmentation is needed.

This act of recursive choosing next best informative attributes is the core idea behind decision tree data science method.

Tree-Structured Models: "Rules"

- No two parents share descendants
- There are no cycles
- The branches always "point downwards"
- Every example always ends up at a leaf node with some specific class determination
  - Probability estimation trees, regression trees (*to be continued..*)

These are some rules that govern the decision tree.

Tree Induction

- How do we create a classification tree from data?
  - **divide-and-conquer** approach
  - take each data subset and ***recursively*** apply attribute selection to find the best attribute to partition it
- When do we stop?
  - The nodes are pure,
  - there are no more variables, or
  - even earlier (over-fitting – *to be continued..*)

Tree Induction: Algorithmic View

Input: labeled input data *D*

Create root node (represents entire *D*)

repeat

Select eligible node for splitting

Split selected node

Choose the "best" attribute for the node

Partition the data according to values of this attribute

Each partition becomes a new node

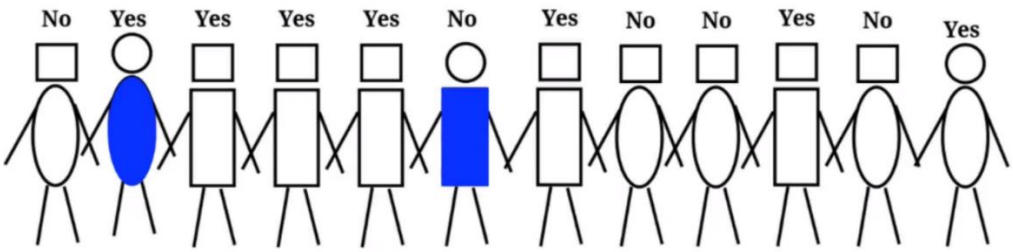
until no more eligible nodes to split

Prune overfitting nodes from the tree

Label each leaf node with its dominant class

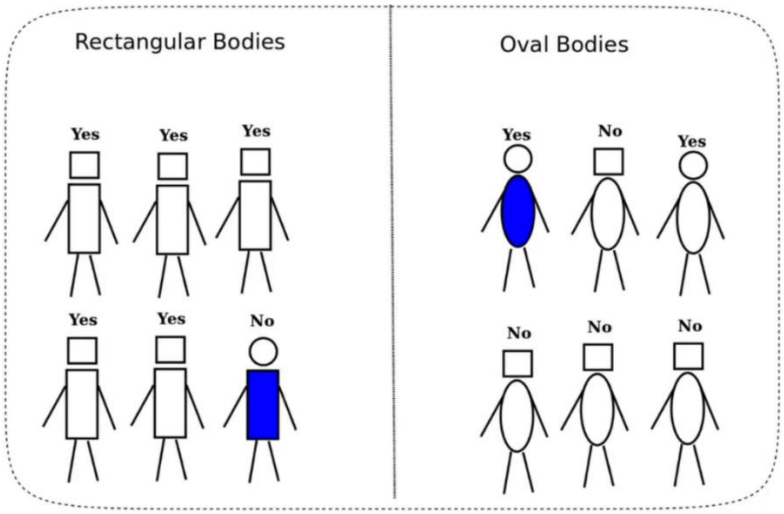
Output: decision tree

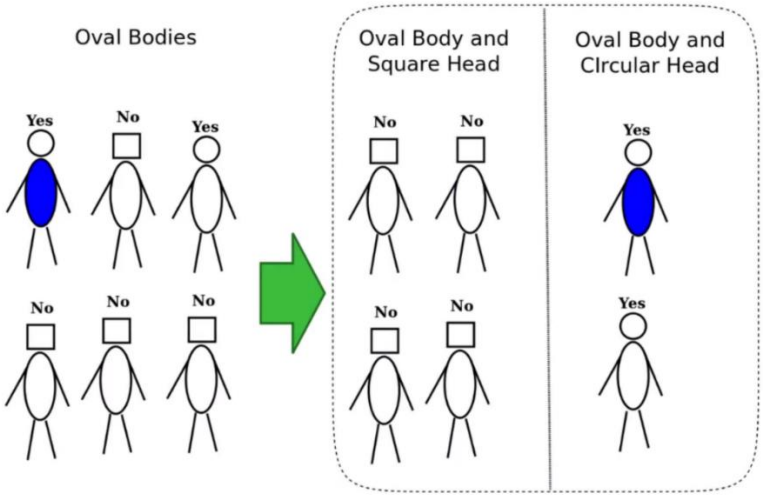
Supervised Segmentation



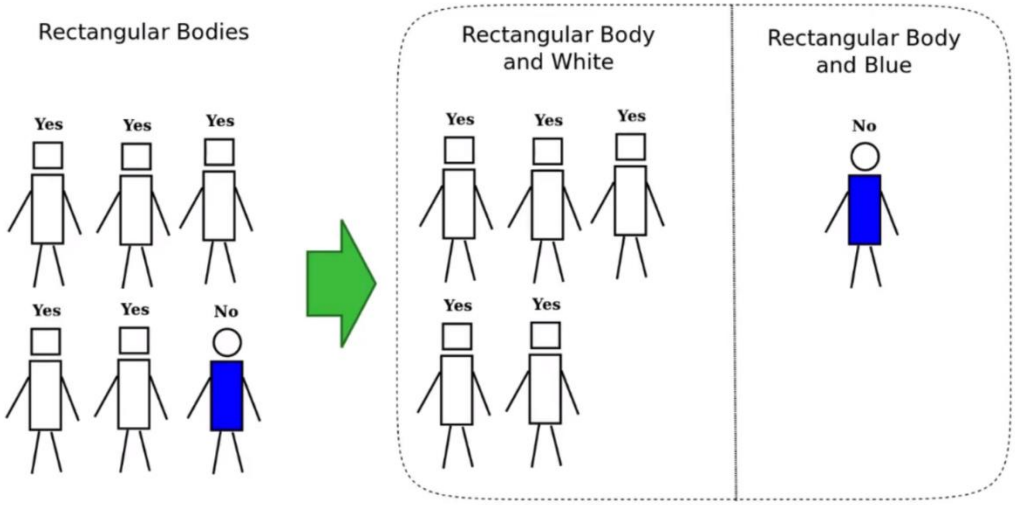
We have several attributes we can use to segment this sample above: Head shape (rectangular or circular), body shape (rectangular or circular), body color (blue or white).

How can we combine this attributes that enables us to make a good prediction? Examining each attribute gives that body shape is the attribute that provides most informative segmentation.

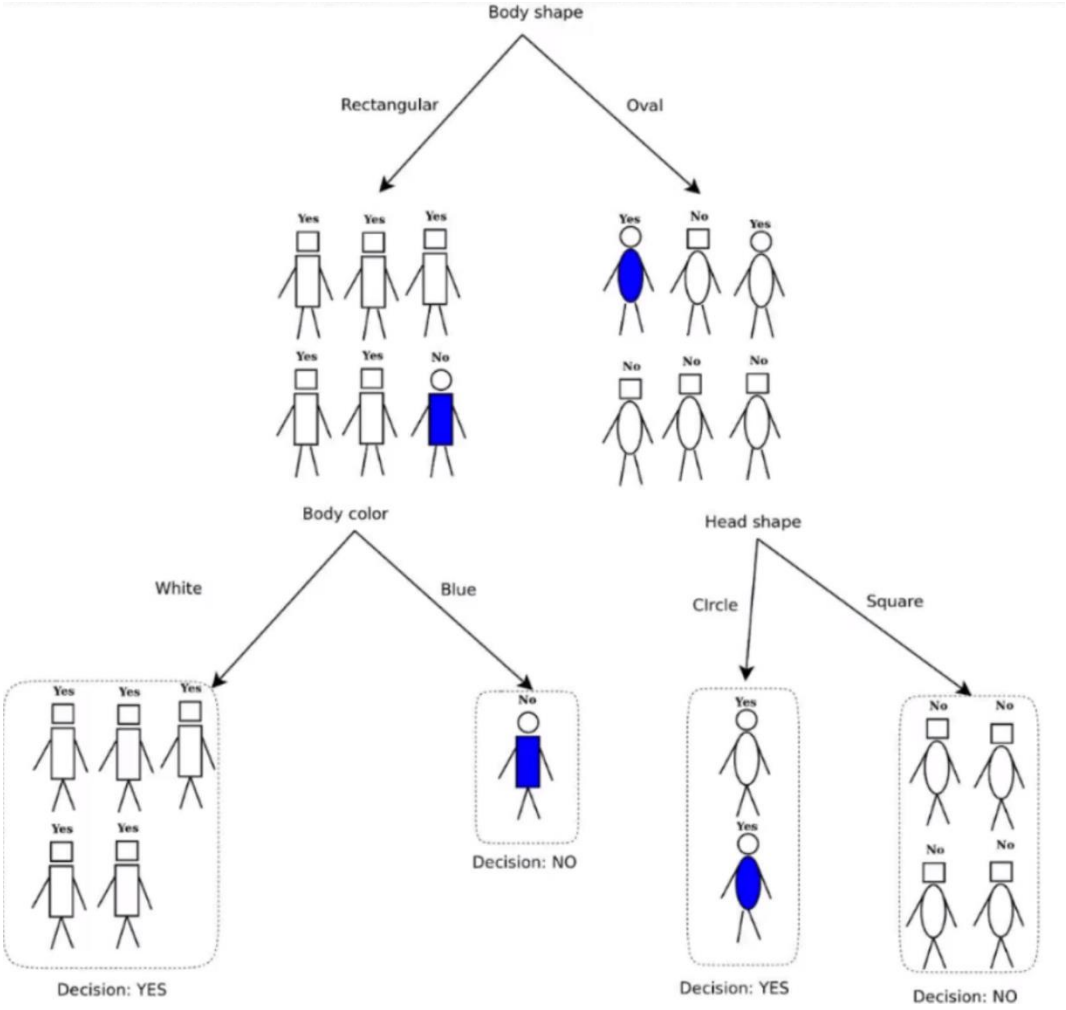




Let's focus on right side of the segmentation, oval bodies, and further break down using the next best informative attribute, head shape.



Next, let's focus on left side of the segmentation, rectangular bodies, and further break down using the next best informative attribute, body color.



Now, let's look at the segmentation we've created. Notice how we recursive choose the next most informative values to create multivariate supervised segmentation.

Question 1

1 / 1 pts

Which component of a decision tree provides the predictions?

Correct!

- leaf nodes

Correct!

A tree's leaf nodes provide the predictions. Leaf nodes are child nodes, but not all child nodes are leaves.

The root node does not make predictions.

The interior nodes are all the nodes between the root node and the leaves.

- interior nodes

interior nodes.

- root node
- child nodes

Question 2

1 / 1 pts

A decision tree can have only two-way splits.

- True

Correct!

- False

Correct! A decision tree can have multi-way splits.

Question 3

1 / 1 pts

Each instance in the data set follows a single path in the decision tree.

Correct!

- True

Correct! Each instance in the data set indeed follows a single path in the decision tree.

- False

Question 4

1 / 1 pts

Decision trees cannot handle numeric attributes when conducting attribute tests in interior nodes.

- True

Correct!

- False

Correct! Decision tree can handle numeric attributes. Specifically, decision tree models simply find a threshold for these numeric attributes that maximizes the information gain of the corresponding attribute.

Question 5

1 / 1 pts

Suppose you build a decision tree to predict whether a customer makes a purchase on the internet (yes or no). A leaf has perfect purity when it contains which of the following?

- Only customers who have made purchases.
- Only customers who have not made purchases.

Correct!

- Either only customers who have made purchases or only customers who have not made purchases.
- An equal number of customers who have made purchases and customers who have not made purchases.

Why Decision Trees?

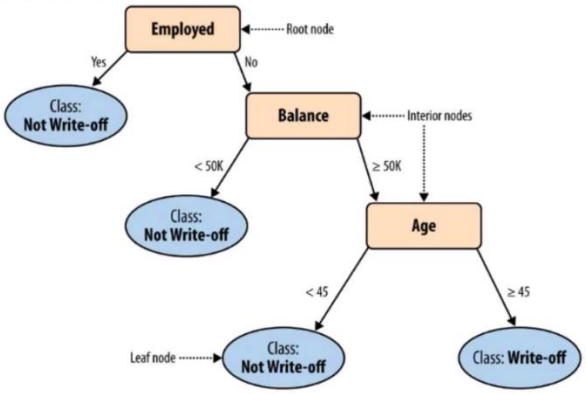
- Decision trees (DTs), or classification trees, are one of the most popular data mining tools
  - (along with linear/logistic regression)
- They're:
  - Easy to understand
  - Easy to implement
  - Easy to use
  - Computationally cheap
- Almost all data mining packages include DTs
- They have advantages for model comprehensibility, which is important for:
  - model evaluation
  - communication to non-DM-savvy stakeholders

Easy to understand (A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility) , easy to implement (divide and conquer)

Trees as Sets of Rules

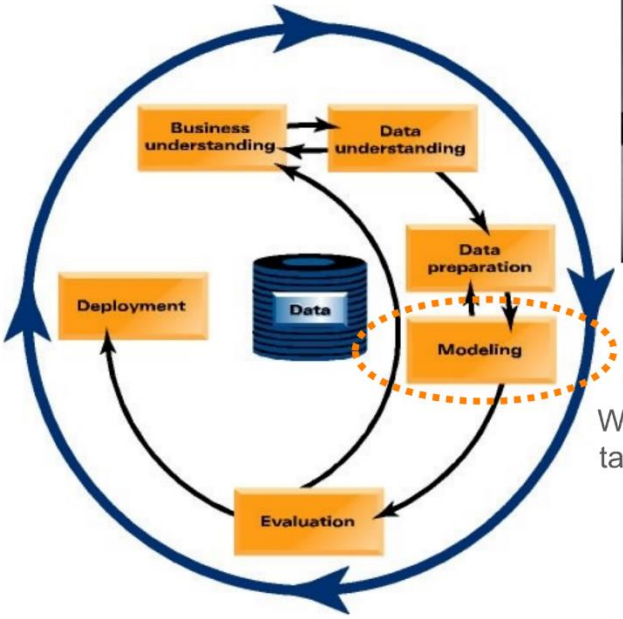
- The classification tree is equivalent to a rule set.
- Each rule consists of the attribute tests along the path connected with **AND**

Trees as Sets of Rules



- IF (Employed = Yes) THEN Class=No Write-off
- IF (Employed = No) AND (Balance < 50k) THEN Class=No Write-off
- IF (Employed = No) AND (Balance ≥ 50k) AND (Age < 45) THEN Class=No Write-off
- IF (Employed = No) AND (Balance ≥ 50k) AND (Age ≥ 45) THEN Class=Write-off

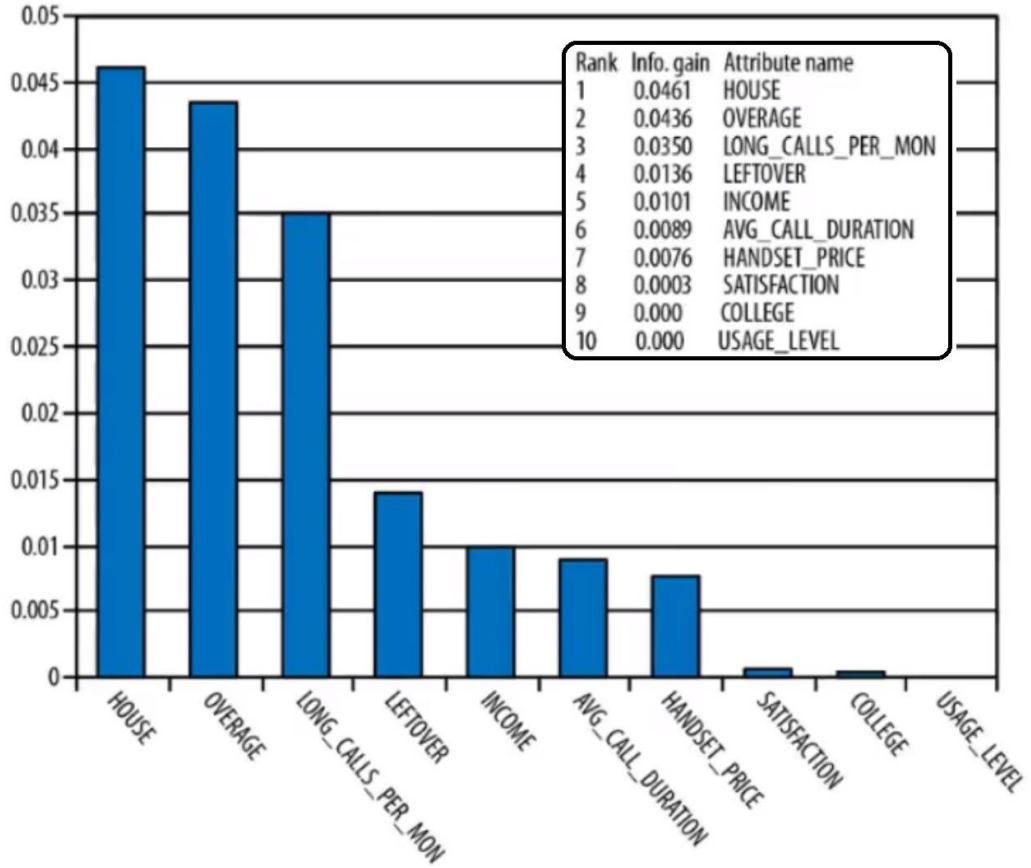
Let's focus back in on actually mining the data..



Which customers should TelCo target with a special offer, prior to contract expiration?

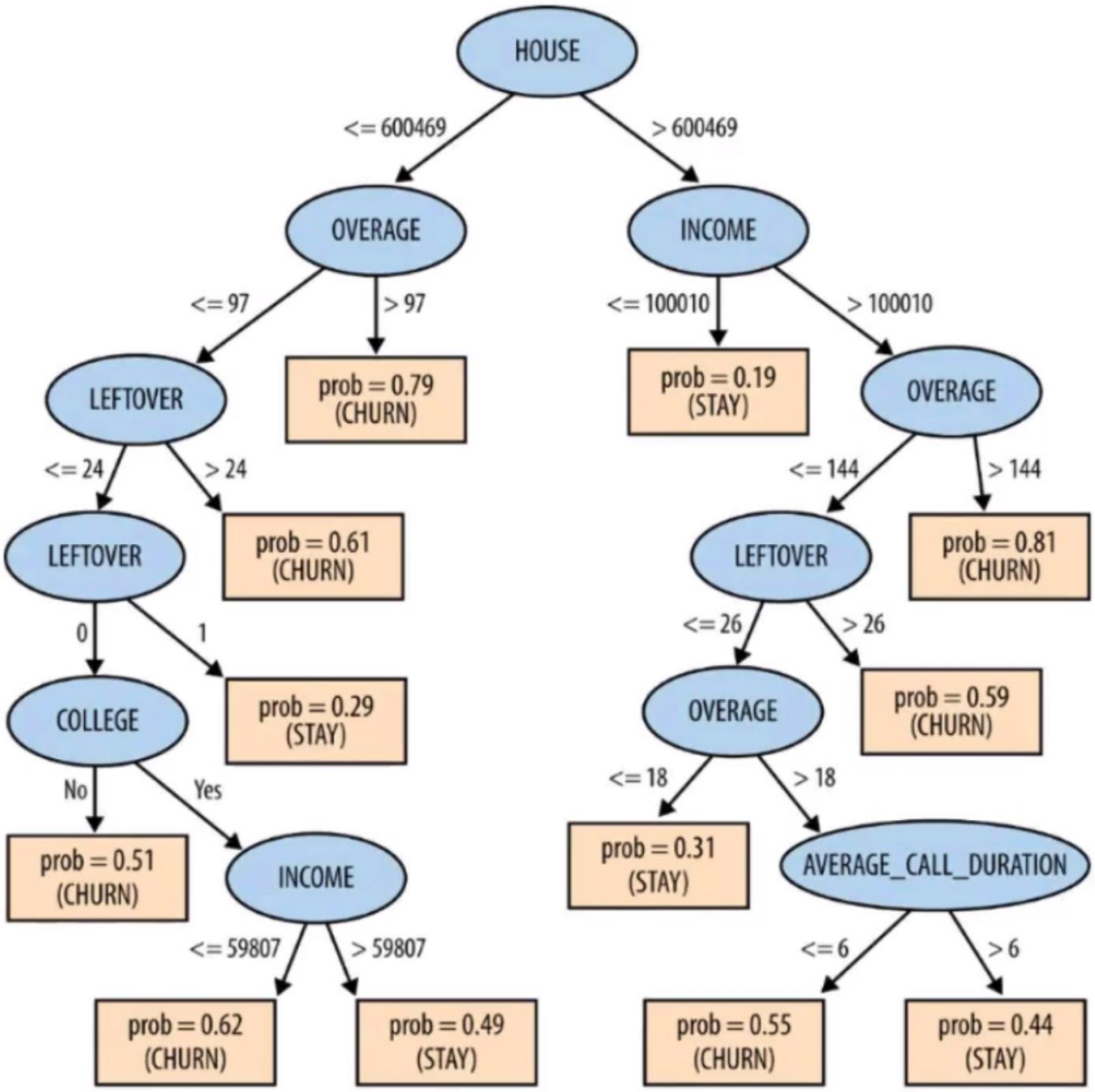
MegaTelCo: Predicting Churn with Tree Induction

Variable	Explanation
COLLEGE	Is the customer college educated?
INCOME	Annual income
OVERAGE	Average overcharges per month
LEFTOVER	Average number of leftover minutes per month
HOUSE	Estimated value of dwelling (from census tract)
HANDSET_PRICE	Cost of phone
LONG_CALLS_PER_MONTH	Average number of long calls (15 mins or over) per month
AVERAGE_CALL_DURATION	Average duration of a call
REPORTED_SATISFACTION	Reported level of satisfaction
REPORTED_USAGE_LEVEL	Self-reported usage level
LEAVE *(Target variable)*	Did the customer stay or leave (churn)?



For the initial population, the most informative value is the attribute, house.

House is numerical variable; thus, we will search all the possible threshold that also maximizes the IG.



House is the most informative attribute for the initial population; besides, once we segment these population into two groups based on the threshold $600,469, we found out overage and income are the next informative attributes to segment these population, respectively.

Every time we segment the population, we have to re-estimate the IG as previous values don't apply.

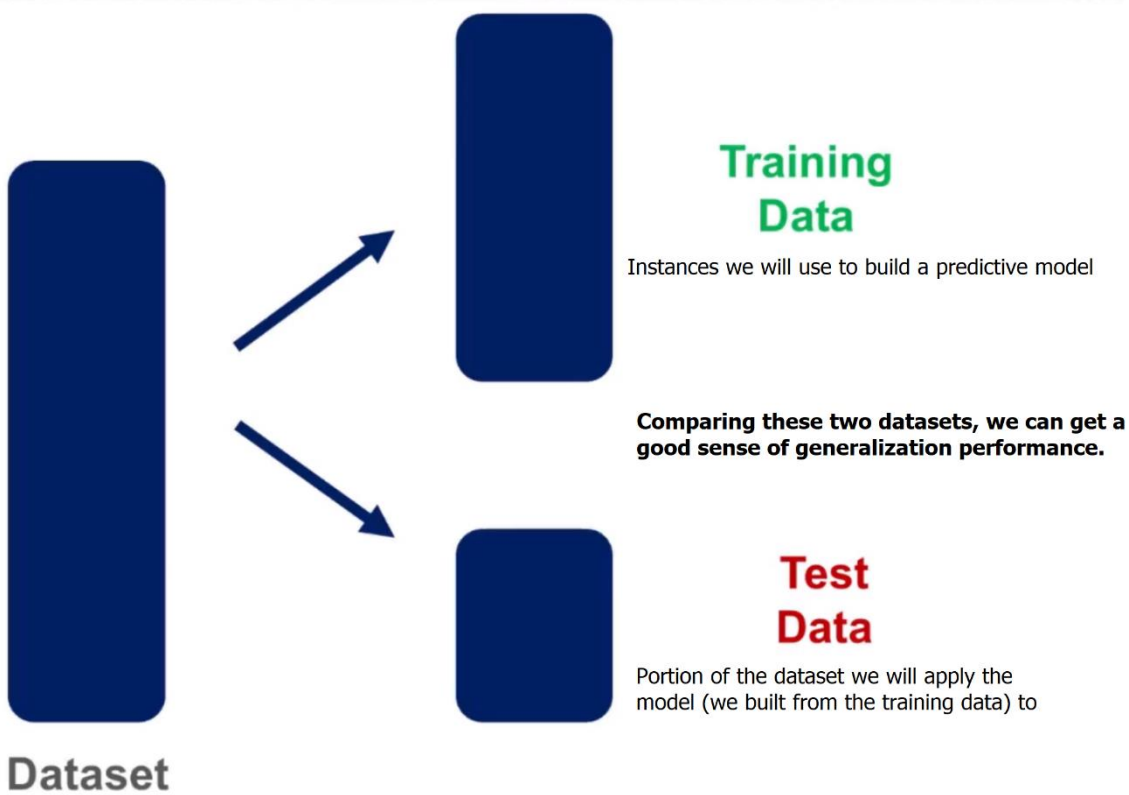
Also keep in mind, we can use the same attributes again (such as 'overage') as long as the decision tree rules are not violated.

Model Evaluation

How do we measure **generalization performance**?

- Multiple classification techniques are available
  - Within the same technique, multiple parameterizations
- Which model is the best model?
  - Need to assess each model's performance
- Reminder:
  - Predictive performance measures must be calculated on **test** (i.e., NOT training!) data

Generalization Performance



Evaluating Classifiers: Plain Accuracy

$$\text{accuracy} = \frac{\text{Number of correct classification decisions made}}{\text{Total number of classification decisions made}}$$

$$= 1 - \text{error rate}$$

- *Too* simplistic..

Accuracy and Misclassification Error

- **Error** = classifying a record as belonging to one class when it belongs to another class
- **Error rate** = percent of misclassified records out of the total records in the validation data
- **Accuracy** = 1 – Error Rate

Misclassifications vs Correct Classifications

Actual Class	Prediction	
Positive	Positive	✓ **True Positive**
Positive	Negative	✗ **False Negative**
		vs
Negative	Positive	✗ **False Positive**
Negative	Negative	✓ **True Negative**

Evaluating Classifiers: The Confusion Matrix

Confusion matrix attempts to avoid the pitfall of simplistic model

- A confusion matrix for a problem involving n classes is an $n \times n$ matrix
  - with the columns labeled with actual classes and the rows labeled with predicted classes
- It separates out the decisions made by the classifier
  - making explicit how one class is being confused for another

Predicted	Actual	
	Positive	**Negative**
Positive	True Positives (TP)	False Positives (FP)
Negative	False Negatives (FN)	True Negatives (TN)

- The errors of the classifier are the false positives and false negatives
  - In statistics:
    - False Positives = Type I errors
    - False Negatives = Type II errors
- Similar type of matrix can be constructed when having more than 2 classes

Type I error: detecting an error that is not present

Type II error: Failing to detect an error that is present

Type I & Type II Errors

False Positive (FP)

False Negative (FP)

Building a Confusion Matrix

Default Truth	Model Prediction
0	0
1	1
0	1
0	1
0	0
1	1
0	0
0	0
1	1
1	0

Actual class Predicted class	Default	No Default	Total
Default	3	2	5
No Default	1	4	5
Total	4	6	10

Incorrect predictions

Predicted	Actual: Positive	Actual: Negative
Positive	True Positives (TP)	False Positives (FP)
Negative	False Negatives (FN)	True Negatives (TN)

Correct predictions

Evaluation of Multi-Class Classification Tasks

- Assume that the target variable has more than two values (classes); e.g., C_1, C_2, ..., C_k
  - E.g., Risk = High, Medium, or Low
- Predicted(C_i)
  - Set of data points predicted to be of class C_i by the model
- Actual(C_i)
  - Set of data points that actually are of class C_i
- Then:

$$\text{Precision}_{C_i} = \frac{|\text{Predicted}(C_i) \cap \text{Actual}(C_i)|}{|\text{Predicted}(C_i)|}$$

The Confusion Matrix & Other Common Metrics

Predicted	Actual: Positive	Actual: Negative
Positive	True Positives (TP)	False Positives (FP)
Negative	False Negatives (FN)	True Negatives (TN)

- Note that all predictions: All = TP + FP + FN + TN
- Some common metrics:
  - **Accuracy** = (TP + TN) / (TP + FP + FN + TN)
  - **Precision**$_{\text{Positive}}$ = TP / (TP + FP) Precision$_{\text{Negative}}$ = TN / (TN + FN)
  - **Recall**$_{\text{Positive}}$ = TP / (TP + FN) Recall$_{\text{Negative}}$ = TN / (TN + FP)
    - Precision and Recall metrics can be calculated for any class
- Very similar calculations when having more than 2 classes

Precision measures **how precise the prediction is.** When we predicted it to be positive, how many instances are actually correct?

Recall measures **number of parameters divided by actual parameters.** Out of the actual true positives, how many of the instances do we recall as being positive?

F-Measure

- The special-purpose combination of the precision and recall score (of an important class):

$$\text{F-measure}_{\text{Positive}} = 2 \times \frac{\text{Precision}_{\text{Positive}} \times \text{Recall}_{\text{Positive}}}{\text{Precision}_{\text{Positive}} + \text{Recall}_{\text{Positive}}}$$

- Represents a *harmonic* mean of precision and recall
- Always between 0 (worst performance) and 1 (best performance)

Naïve (or Majority) Rule

- **Naïve rule**: classify all records as belonging to the majority (most prevalent) class
  - Not a real method (typically does not make sense in many real-world applications)
  - Sometimes used as benchmark: we hope to outperform it
  - However, when goal is to identify high-value but rare outcomes, we may do well by doing worse than the naïve rule
    - to be continued..

Predicted	Actual	
	Positive	**Negative**
Positive	True Positives (TP)	False Positives (FP)
Negative	False Negatives (FN)	True Negatives (TN)

(REVIEW) Naive Bayes classifier assumes that the presence of a particular feature in a class is un-related to the presence of any other feature.

Incorrect

Question 21

0 / 5 pts

A predictive model has been applied to a test dataset and has classified 87 records as fraudulent (31 correctly so) and 953 as non-fraudulent (919 correctly so).

What is the accuracy rate that would be achieved by naïve (majority) rule on this data?

0.884

	P	N
	31	56
	34	919

$$\frac{975}{65+975}$$

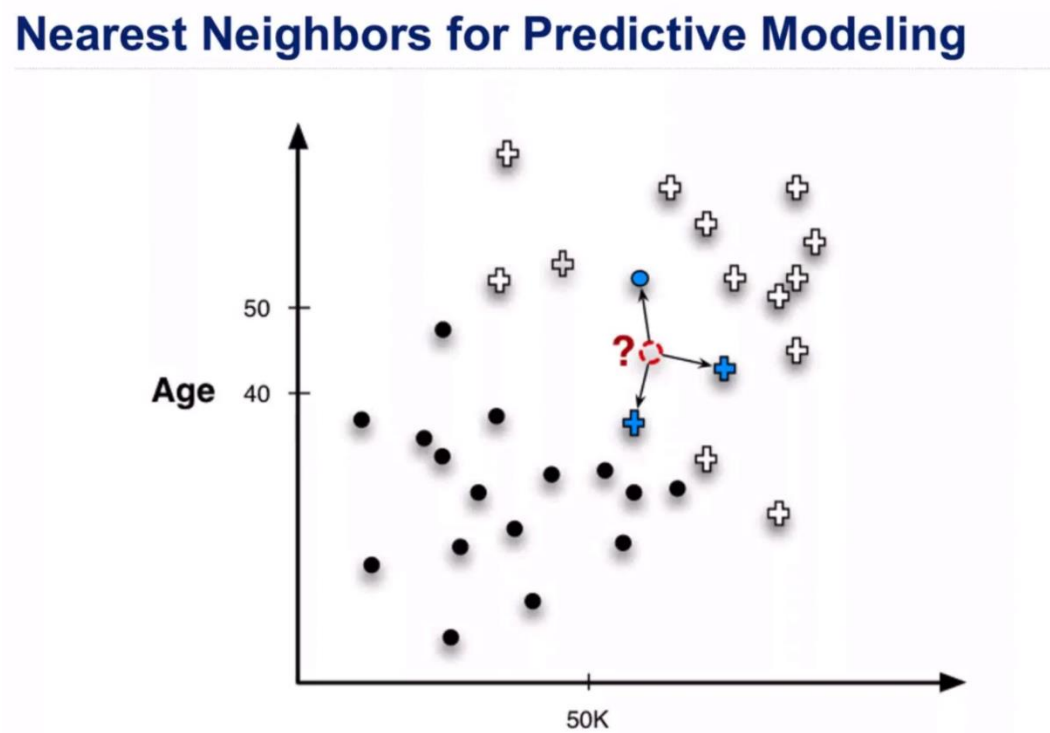
k-Nearest Neighbors (kNN) algorithm

kNN Assumption: points that are close to one another are similar

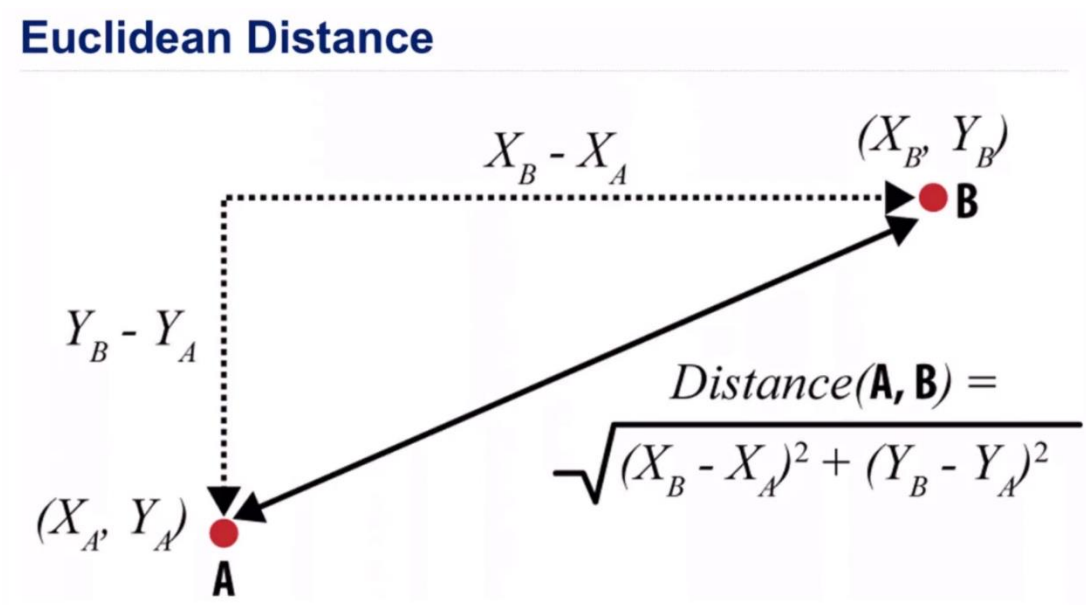
Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point.

k-NN Classifiers: Basic Idea

- k-NN stands for "k Nearest Neighbors"
- For a given p-dimensional record to be classified, **identify nearby records**
  - "Near" means records with similar attribute values $x_1, x_2, \ldots, x_p$
- Classify the record as whatever the predominant class is among the nearby records (the "neighbors")



There are three variables near the target of interests, of which two of them are in positive class. Therefore, we can make the prediction that target of the interest also belong to the positive class.



We can determine the distance between target of interest and variable using the formula above.

Similarity and Distance

- If two *objects* can be represented as *feature vectors*, then we can compute the distance between them
- When retrieving neighbors, we don't use the target variable as an attribute

Attribute	Person A	Person B
Age	23	40
Years at current address	2	10
Residential status (1=Owner, 2=Renter, 3=Other)	2	1

Residential status is categorical variable unlike the first two numeric variables, so we need to somehow find ways to convert this to numeric to make all the attribute in the dataset homogenous. And we can do this by **setting a threshold that provides binary segmentation as a result.**

Nearest Neighbors for Predictive Modeling

Customer	Age	Income (1000s)	Cards	Response (target)	Distance from David
David	*37*	*50*	*2*	*?* **Yes**	0
John	35	35	3	Yes	$\sqrt{(35-37)^2+(35-50)^2+(3-2)^2} = \boxed{15.16}$
Rachael	22	50	2	No	$\sqrt{(22-37)^2+(50-50)^2+(2-2)^2} = \boxed{15}$
Ruth	63	200	1	No	$\sqrt{(63-37)^2+(200-50)^2+(1-2)^2} = 152.23$
Jefferson	59	170	1	No	$\sqrt{(59-37)^2+(170-50)^2+(1-2)^2} = 122$
Norah	25	40	4	Yes	$\sqrt{(25-37)^2+(40-50)^2+(4-2)^2} = \boxed{15.74}$

Two out of three closest values belong to the 'Yes' for their response.

Notice the range of the attribute for age (63-22) is widely different from that of income (200-35).

Other Distance Functions

$$d_{Manhattan}(\boldsymbol{X},\boldsymbol{Y}) = \|\boldsymbol{X}-\boldsymbol{Y}\|_1 = |x_1 - y_1| + |x_2 - y_2| + \cdots$$

$$d_{Jaccard}(X,Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|} \qquad d_{Cosine}(\boldsymbol{X},\boldsymbol{Y}) = 1 - \frac{\boldsymbol{X}\cdot\boldsymbol{Y}}{\|\boldsymbol{X}\|_2 \cdot \|\boldsymbol{Y}\|_2}$$

Key Issue: Normalization

- **Problem**: Raw distance measures are highly influenced by scale of measurements
  - E.g., income=$100,000; height=1.50m
    - Income will highly dominate height in distance computations, because of highly differing scales of the two attributes
- **Solution**: normalize (standardize) the data first
  - Normalization transforms variables (attributes) of different scales (different orders of magnitude) into similar scale so that they can be compared and can contribute equally to the distance computations

Normalization Approaches

- Scaling/normalizing is used to standardize the intervals before measuring distances
  - min-max scaling numerical attribute to interval [0,1]

$$z = (x - \min) / (\max - \min)$$

  - x := original value of the attribute
  - min := smallest value of the attribute
  - max := largest value of the attribute
  - z is the resulting (scaled) value of the attribute in the range [0,1]
- z-score scaling to standardize the intervals

$$z = (x - m) / s$$

  - m := mean value of the attribute
  - s := standard deviation (or mean absolute deviation, which is more robust to outliers than standard deviation)
- When attributes have different importance
  - Weighted distances may be used
  - E.g., $d(A, B) = w_1|a_1 - b_1| + \cdots + w_k|a_k - b_k|$

Min-Max Approach: Example

- Setting: Consider the age and income data of several employees (as provided below)
- Task: Normalize the data using min-max approach

$$z = (x - \min) / (\max - \min)$$

Name	Age	Income
Alice	70	100,000
Bob	25	50,000
Cindy	30	60,000
David	20	70,000
Earl	60	80,000

Name	Age(Norm)	Income(Norm)
Alice	1.0	1.0
Bob	0.1	0.0
Cindy	0.2	0.2
David	0.0	0.4
Earl	0.8	0.6

Maximum value becomes 1 and minimum values becomes 0.

How many neighbors?

- No simple answer
- ***Odd numbers*** are convenient for breaking ties for majority vote classification with two-class problems
- In general, the greater k is the more the estimates are smoothed out among neighbors
  - What happens if k=n where n is the size of the training data set?
- We will revisit this question again (To be Continued...)

Greater k having smooth out among neighbors are closely related to the 'overfitting' concept.

How much influence?

- Even when we have chosen k, some "nearest neighbors" are nearer than others
- Should this influence how they're used?
  - i.e., Should we allow the most similar neighbors to have greater influence on our prediction?
- We can use *weighted voting* or *similarity moderated voting* such that each neighbor's contribution is scaled by its similarity.
- The inverse of the square of the distance is commonly used e.g., $w(x, y) = \frac{1}{dist^2(x,y)}$

Similarity-moderated kNN

Name	Distance	Similarity weight	Contribution	Class
Rachael	15.0	0.004444	0.344	No
John	15.2	0.004348	0.336	Yes
Norah	15.7	0.004032	0.312	Yes
Jefferson	122.0	0.000067	0.005	No
Ruth	152.2	0.000043	0.003	No
		0.012934		

To calculate 'Similarity weight', simply take the **inverse of the square values of the distance**.

0.012934 is the sum of the all values in the 'Similarity weight' column.

Contribution is calculated by dividing each weight from the sum (e.g. 0.004444/0.012934 = 0.344)

We can also add up all the contribution term by the class and find that Yes is 0.65 while No is 0.35.

0.65 Yes
0.35 No

Advantage: Weighted scoring reduces the importance of deciding how many neighbors to use.

Question 1

1 / 1 pts

Each attribute can appear only in a single split in the decision tree.

- True
- False

Correct!

Correct! As long as the rules of the decision tree are not violated, then, an attribute can be used in multiple splits in the decision tree. Recall, for instance, the attribute OVERAGE in the telecommunications company example was used in multiple splits in the decision tree.

Question 2

0 / 1 pts

Decision trees can only handle categorical variables as attributes in interior nodes.

You Answered

- True

Decision trees can handle numeric variables as attributes in interior nodes. Recall, for instance, we used the account balance, which is a numeric attribute, as an interior node in a decision tree.

Correct Answer

- False

Question 3

1 / 1 pts

In order to estimate the generalization performance of a decision tree model, I estimate the performance using:

- The training set
- Correct! The test set

  Correct! The generalization performance is estimated using data points the model has not seen (i.e., using the test set).
- Either the training set or the test set

Question 4

1 / 1 pts

When using a similarity-moderated kNN to make predictions for two different data points, even if I always use all the instances in the data to generate predictions, I will not always generate the same prediction.

- Correct! True

  Correct! The weights will allow the most similar entities to have a higher influence on my predictions and, therefore, I will not always generate the same prediction.
- False

Question 5

1 / 1 pts

The accuracy metric is able to discriminate between the different types of correct classifications a classifier makes.

- True
- Correct! False

Correct! The accuracy metric is not able to discriminate between the different types of correct classifications a classifier makes. It treats True Positives and True Negatives as simply correct decisions.

Question 9

0 / 5 pts

You have built a 5NN model after you've normalized the features based on the z-score transformation technique. Suppose you build the same 5NN but this time you change the normalization technique and you use min-max scaling instead of the z-score transformation. Will the predictions the second model generates change compared to the prediction of the first model?

- You Answered: No, the predictions are not likely to change
- Correct Answer: Yes, the predictions are likely to change

Yes, different method can generate different prediction.