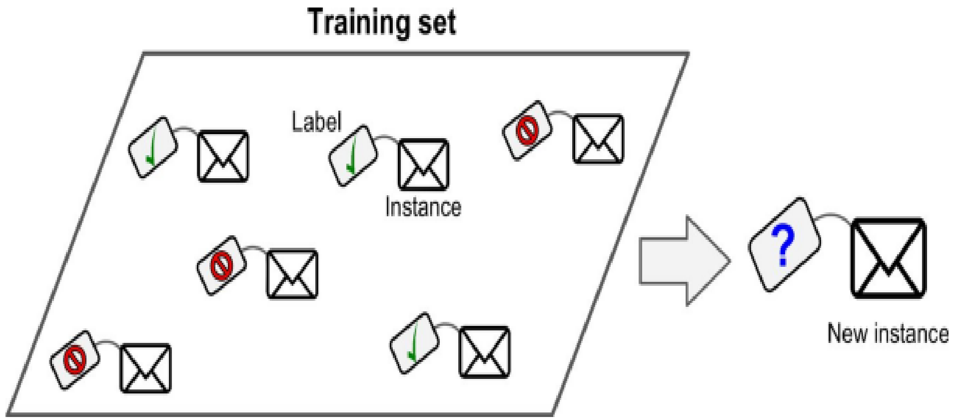
Linear regression & Decision tree algorithm

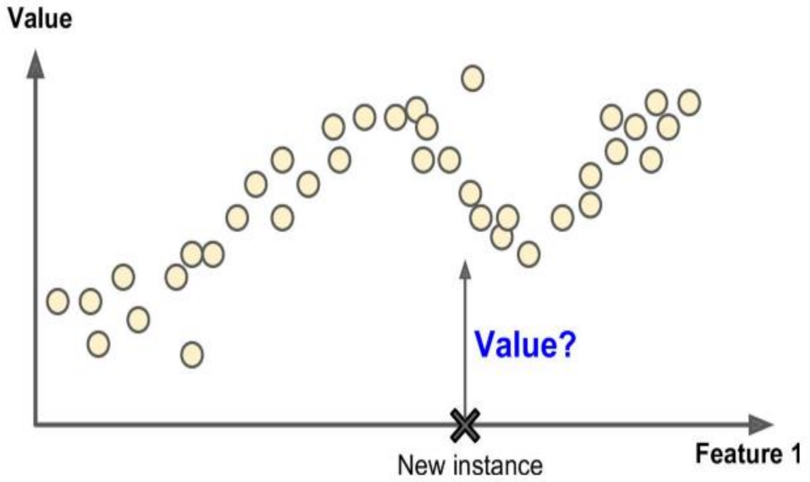
Module four

Supervised learning

- There is label field in the training dataset
- Examples are:
  - KNN
  - Decision tree
  - Random forest
  - Regression
  - Logistic regression
  - ANN
  - Deep Neural network
  - SVM



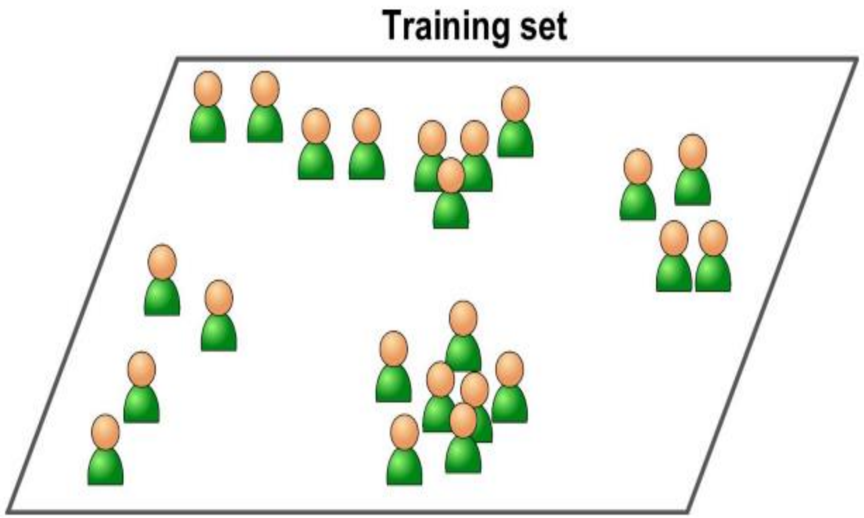
Classification techniques



Regression techniques

Unsupervised learning-1/2

- There are no labels assigned in the training dataset. The system tries to learn within a teacher.
- Examples are:
  - Clustering techniques (e.g. KMeans/Hierarchical clustering analysis)
  - Principle components analysis(PCA)
  - Association
  - Anomaly detection



An unlabeled training dataset for unsupervised learning

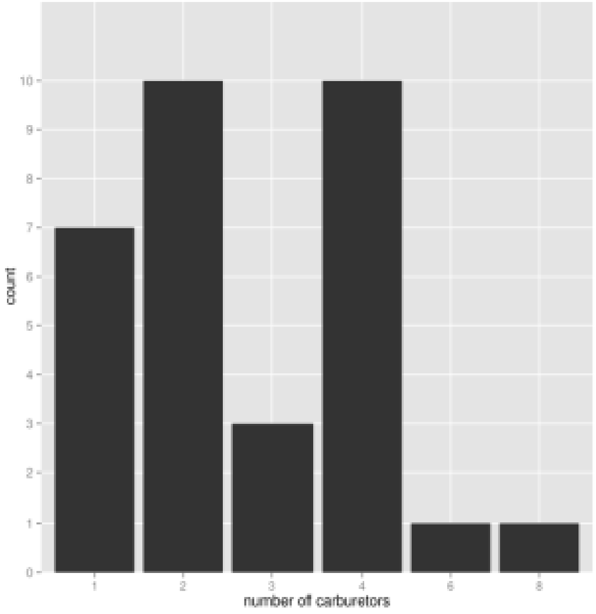
Basic statistics, linear regression and correlation

Agenda

- Describe Univariate data
  - Central tendency
  - Spread
  - Distribution
- Describe multivariate data
  - Linear regression
  - Correlation

Univariate data

- Categorical data
  - Nominal
  - Ordinal (categorical variables that can be sorted or ordered. E.g. t-shirt size)
- Continuous data
  - Continuous variables can be discretized to become categorical data
- Frequency distributions



Central tendency

- Categorical data
  - Mode
- Continuous data
  - Mean
  - Median

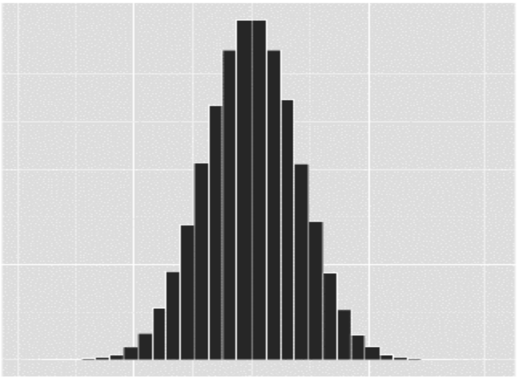


Figure 2.3: A normal distribution

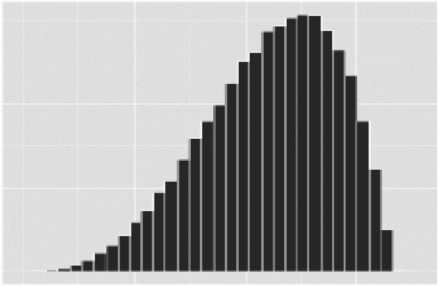


Figure 2.4a: A negatively skewed distribution

Degree of skewness

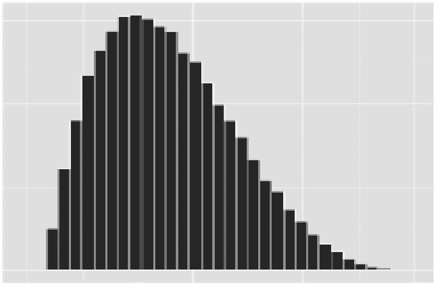


Figure 2.4b: A positively skewed distribution

Spread

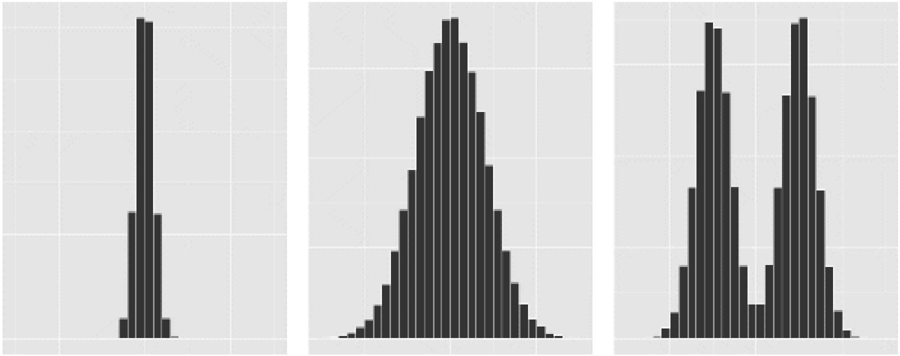


Figure 2.5: three distributions with the same mean and median

- Variance
- Standard deviation

$$\sum_{i=1}^{n} \frac{(x_i - \mu)^2}{n} \equiv \sigma^2,$$

$$\sum_{i=1}^{n} \frac{(x_i - \mu)^2}{n-1} \equiv s^2,$$

$$\sqrt{\sum_{i=1}^{n} \frac{(x_i - \mu)^2}{n}} \equiv \sigma$$

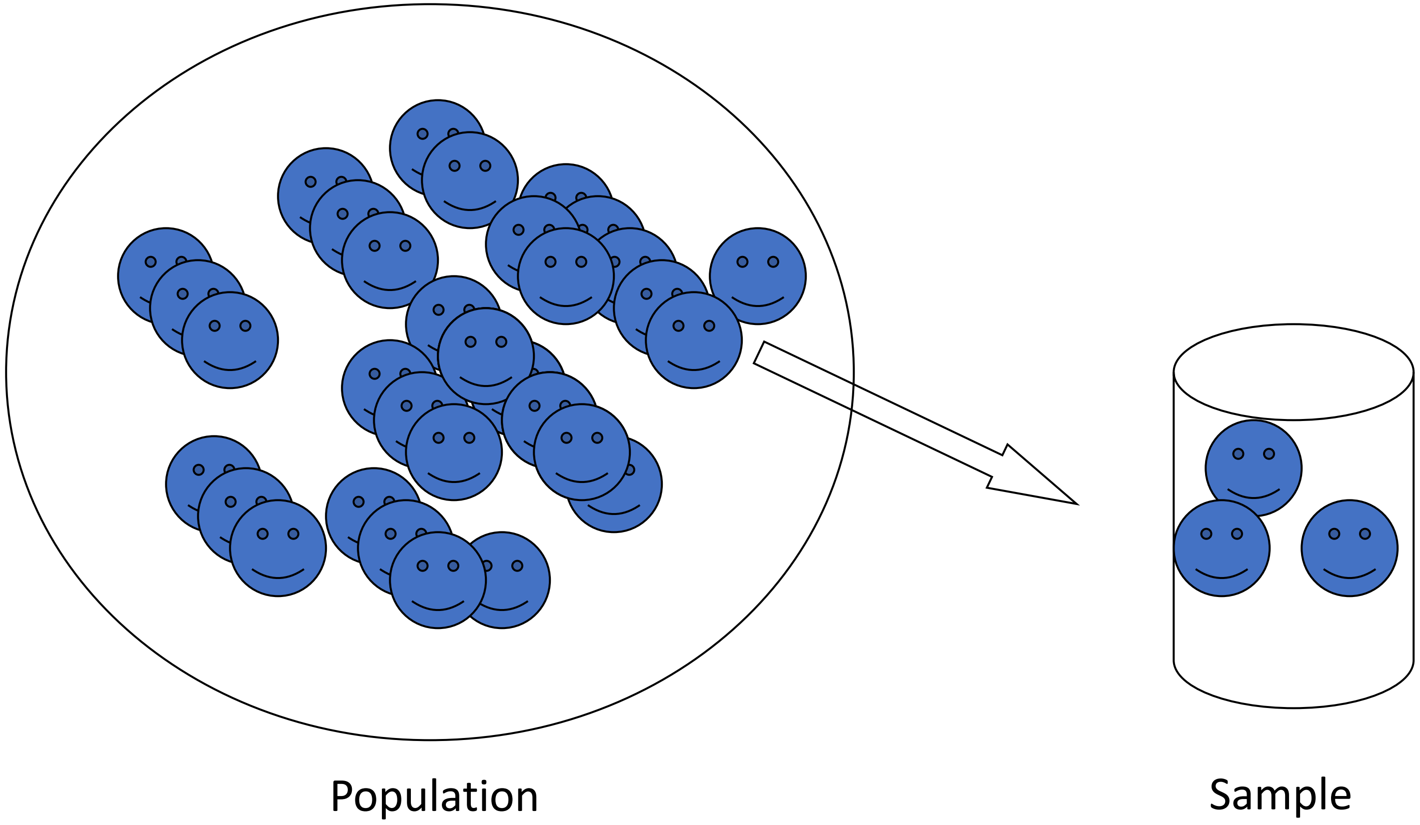
Degrees of freedom

$$\sqrt{\sum_{i=1}^{n} \frac{(x_i - \mu)^2}{n-1}} \equiv s$$

SD of a sample

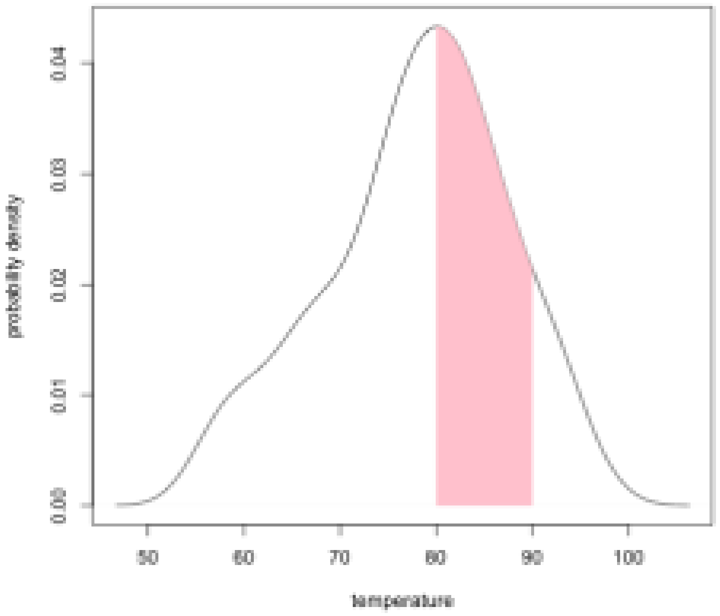
Populations,samples and estimation

- One of the core ideas of statistics is that we can use a subset of a group, study it and then make inferences or conclusions about that much larger group.



Probability density function (PDF)

- In probability theory, a **probability density function** (**PDF**), or **density** of a continuous random variable, is a function that describes the relative likelihood for this random variable to take on a given value. The probability of the random variable falling within a particular range of values is given by the integral of this variable's density over that range—that is, it is given by the area under the density function but above the horizontal axis and between the lowest and greatest values of the range. The probability density function is nonnegative everywhere, and its integral over the entire space is equal to one.



Descriptive Statistics

Summarizing Data:

- ✓ Central Tendency (or Groups’ “Middle Values”)
  - ✓ Mean
  - ✓ Median
  - ✓ Mode
- ✓ Variation (or Summary of Differences Within Groups)
  - ✓ Range
  - ✓ Interquartile Range
  - ✓ Variance
  - ✓ Standard Deviation
- …Wait! There’s more

Box-Plots

A way to graphically portray almost all the descriptive statistics at once is the box-plot.

A box-plot shows: Upper and lower quartiles

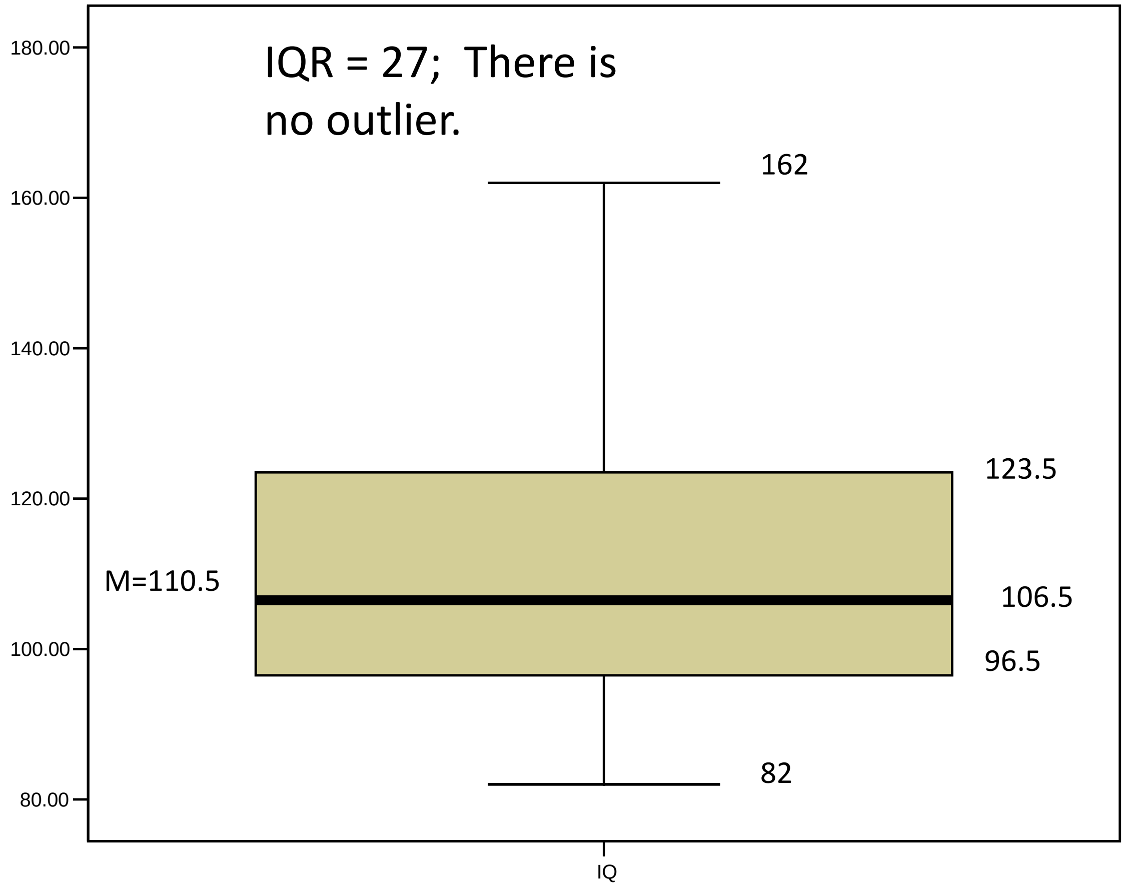
Mean

Median

Range

Outliers (1.5 IQR)

Example: Box-Plots



Multivariate data

- To determine the relationship among different variables
- The focus in this chapter is the relationship among continuous varaiables

Topics Covered:

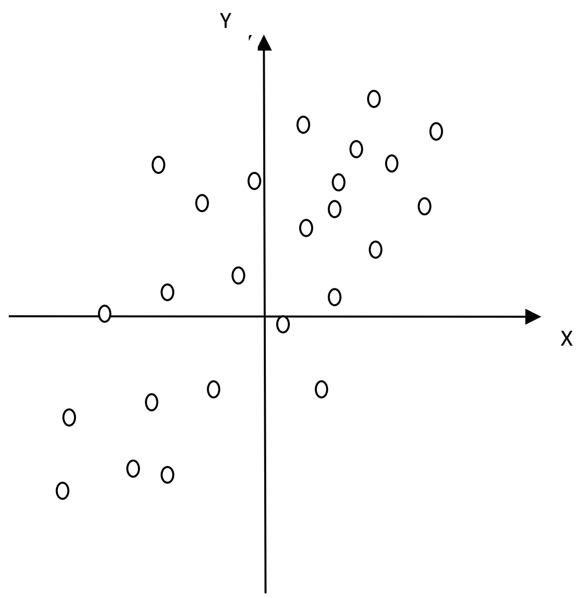
- Is there a relationship between x and y?
- What is the strength of this relationship
  - Pearson's r
- Can we describe this relationship and use this to predict y from x?
  - Regression

The relationship between *x* and *y*

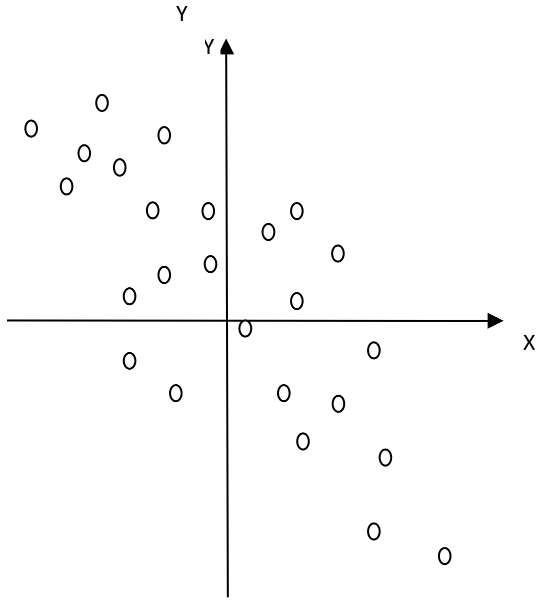
- Correlation: is there a relationship between 2 variables?
- Regression: how well a certain independent variable predict dependent variable?

- CORRELATION ≠ CAUSATION
  - In order to infer causality: manipulate independent variable and observe effect on dependent variable
  - For example, there may be a strong association between mortality and time per day spent watching movies, but before doctors should start recommending that we all should match more movies, we need to rule out another explanation- younger people watch more movies and are less likely to die.

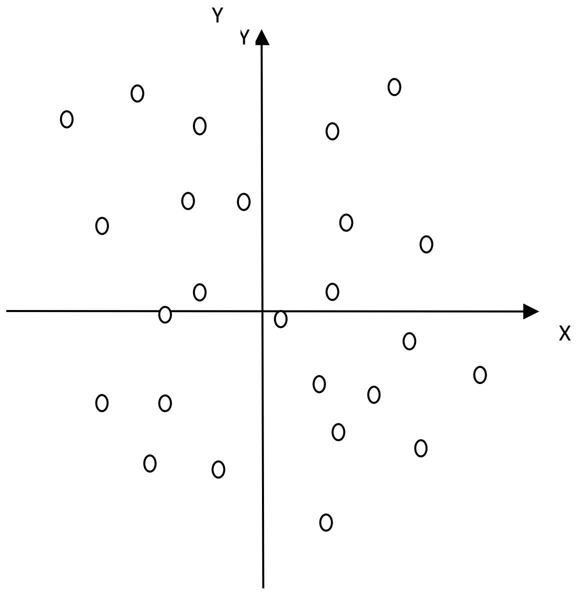
Scattergrams



Positive correlation



Negative correlation



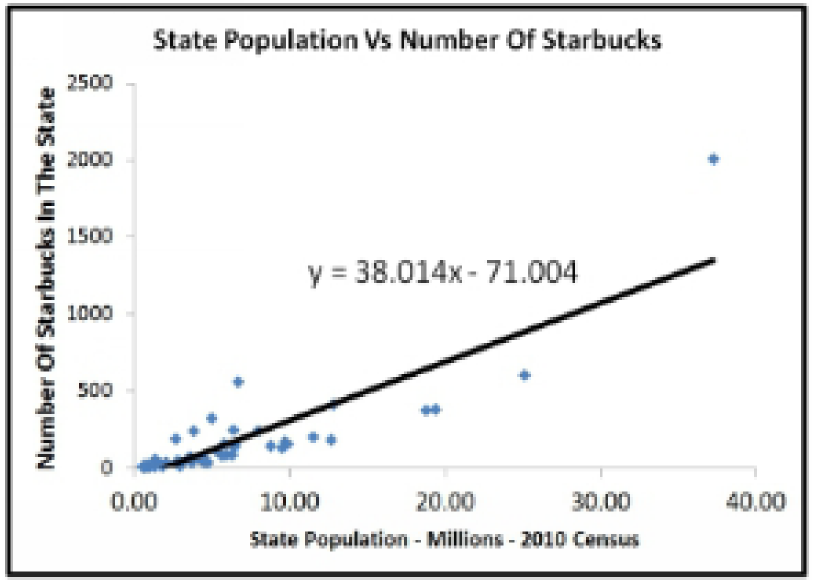
No correlation

What is linear regression

- Linear regression is a way of predicting an unknown variable using results that you do know.
- If you have a set of x and y values, you can use a regression equation to make a straight line relating the x and y.
- The reason you might want to do this is if you know some information, and want to estimate other information.
- For instance, you might have measured the fuel economy in your car when you were driving 30 miles per hour, when you were driving 40 miles per hour, and when you were driving 75 miles per hour.
- Now you are planning a cross country road trip and plan to average 60 miles per hour, and want to estimate what fuel economy you will have so that you can budget how much money you will need for gas.

Example

- The chart on the right shows an example of linear regression using real world data.
- It shows the relationship between the population of states within the United States, and the number of Starbucks (a coffee chain restaurant) within that state.



Equation of the linear regression

Constants

$$y = a + bx$$

Variables

Target attribute

independent attribute

$$y = a + b_1x_1 + b_2x_2 + b_3x_3 + \cdots$$

R square-A way to evaluate the regression-1/2

Sum Squared Regression Error

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}}$$

Sum Squared Total Error

Sum Over All The Data Points

Square The Result

$$SS_{Total} = \sum (y_i - \bar{y})^2$$

Sum Squared Total Error

Each Data Point

Mean Value

Sum Over All The Data Points

Square The Result

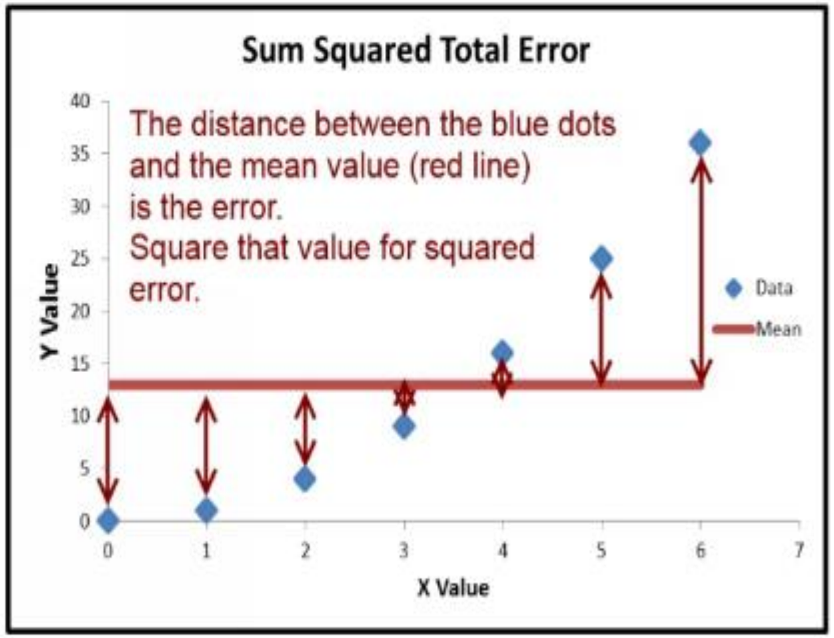
$$SS_{Regression} = \sum (y_i - y_{Regression})^2$$

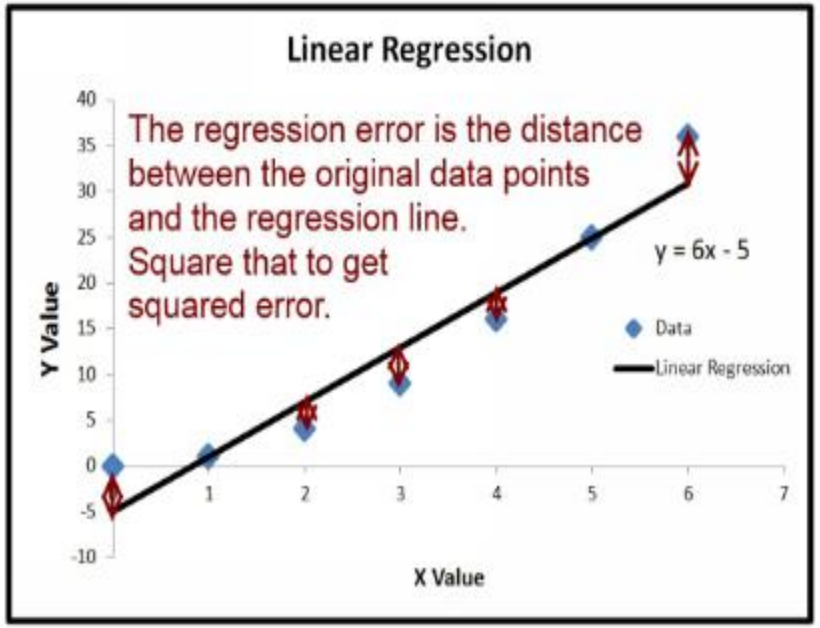
Sum Squared Regression Error

Each Data Point

Regression Value

R square-A way to evaluate the regression-2/2

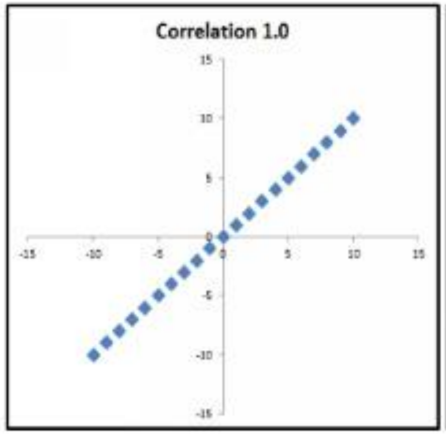


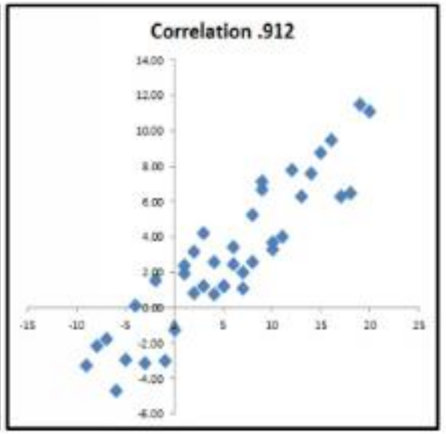


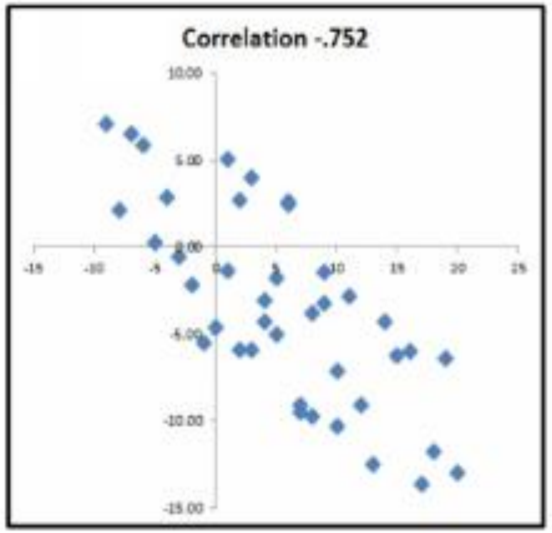
Correlation

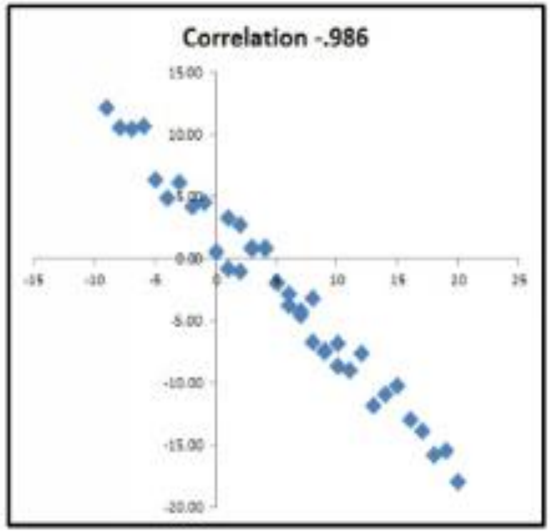
- Correlation is a measure of how closely two variables move together.
- Pearson’s correlation coefficient is a common measure of correlation, and it ranges from +1 for two variables that are perfectly in sync with each other, to 0 when they have no correlation, to -1 when the two variables are moving opposite to each other.

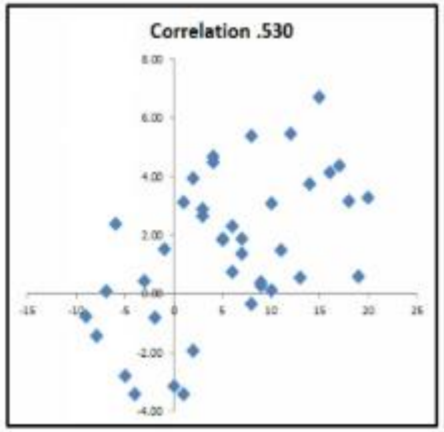
Example: Correlation

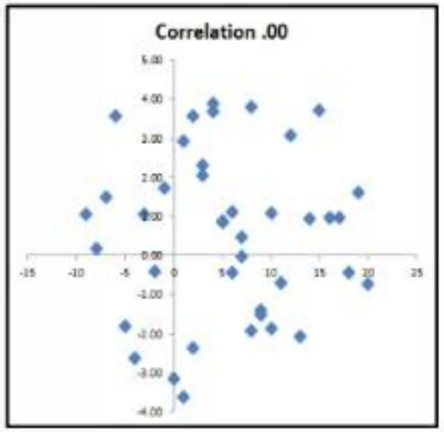




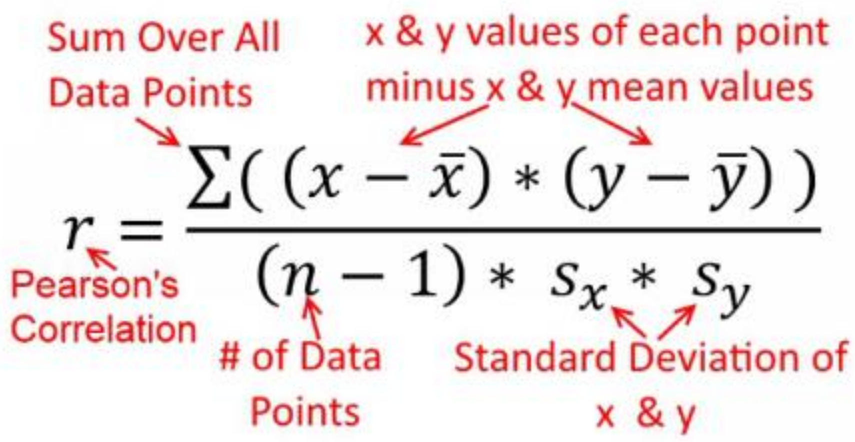






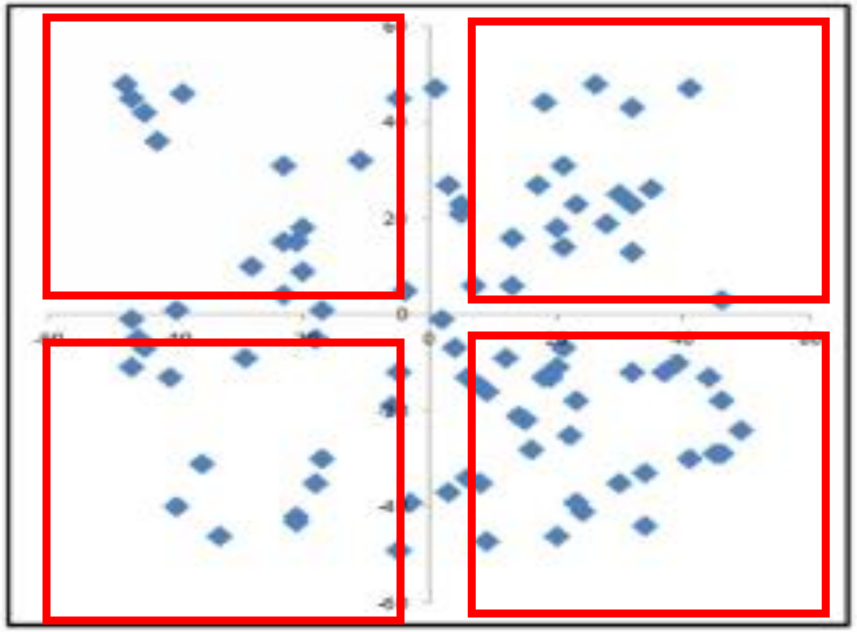


Pearson’s correlation coefficient



Quadrant Four:
(x-mean(x))*(y-mean(y)) negative

Quadrant One:
(x-mean(x))*(y-mean(y)) positive



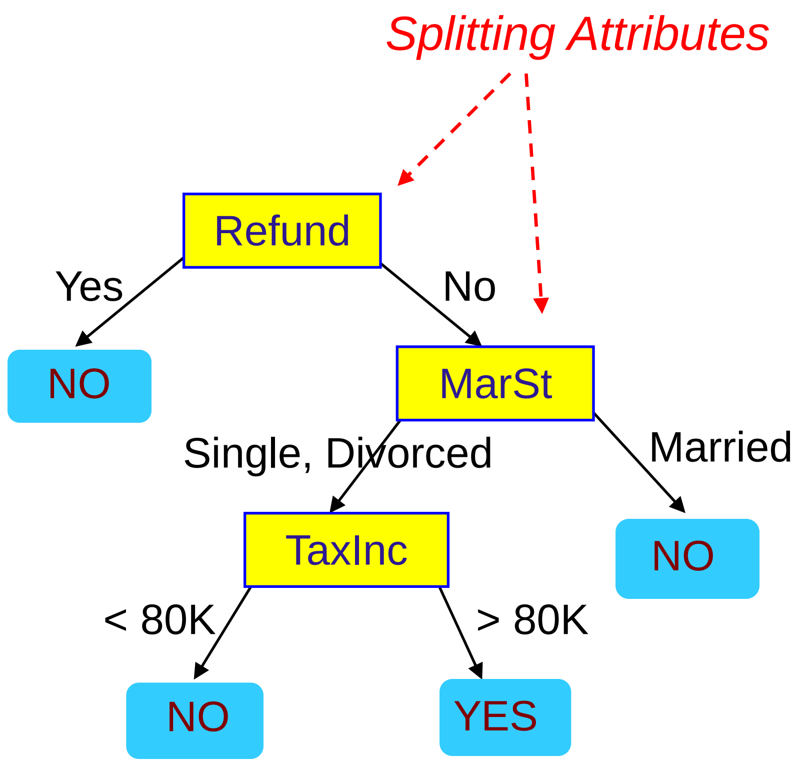
Quadrant Three:
(x-mean(x))*(y-mean(y)) positive

Quadrant Two:
(x-mean(x))*(y-mean(y)) negative

Example of a Decision Tree

		categorical	categorical	continuous	class
	Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
	1	Yes	Single	125K	**No**
	2	No	Married	100K	**No**
	3	No	Single	70K	**No**
	4	Yes	Married	120K	**No**
	5	No	Divorced	95K	**Yes**
	6	No	Married	60K	**No**
	7	Yes	Divorced	220K	**No**
	8	No	Single	85K	**Yes**
	9	No	Married	75K	**No**
	10	No	Single	90K	**Yes**

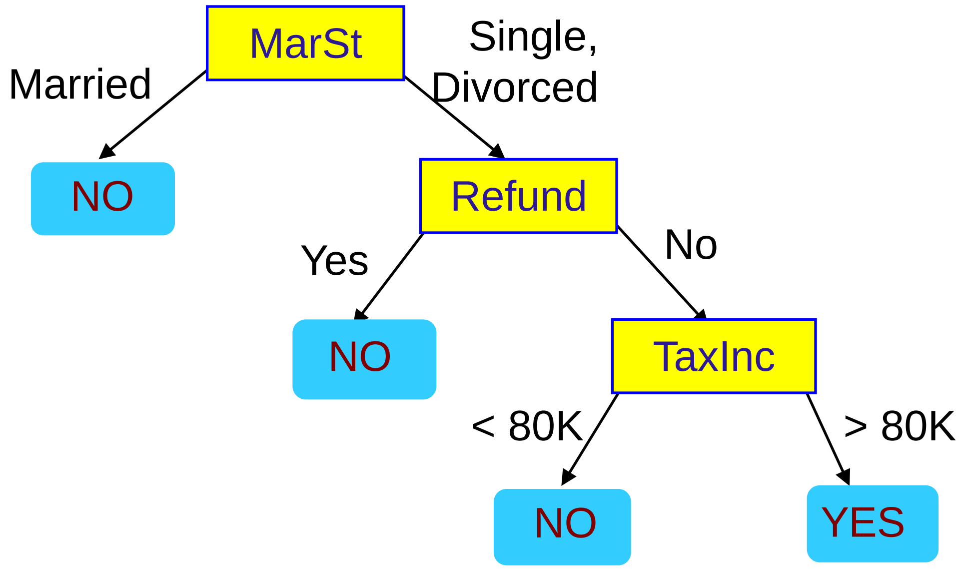
Training Data



Model: Decision Tree

Another Example of Decision Tree

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**



There could be more than one tree that fits the same data!

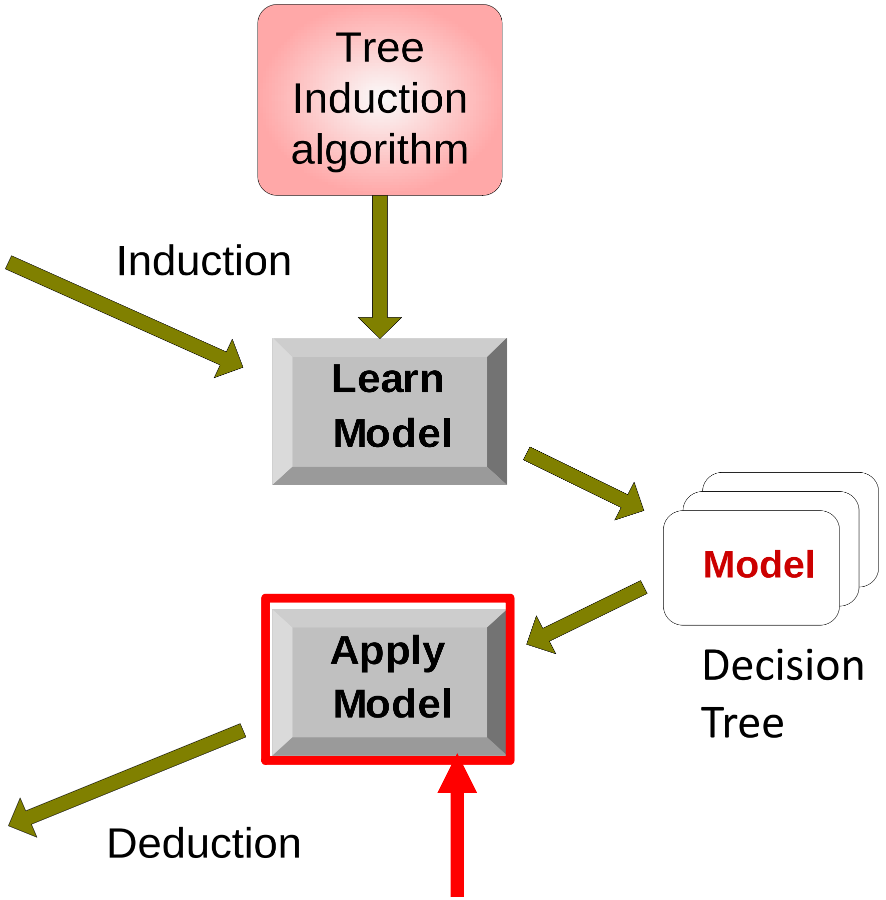
Decision Tree Classification Task

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

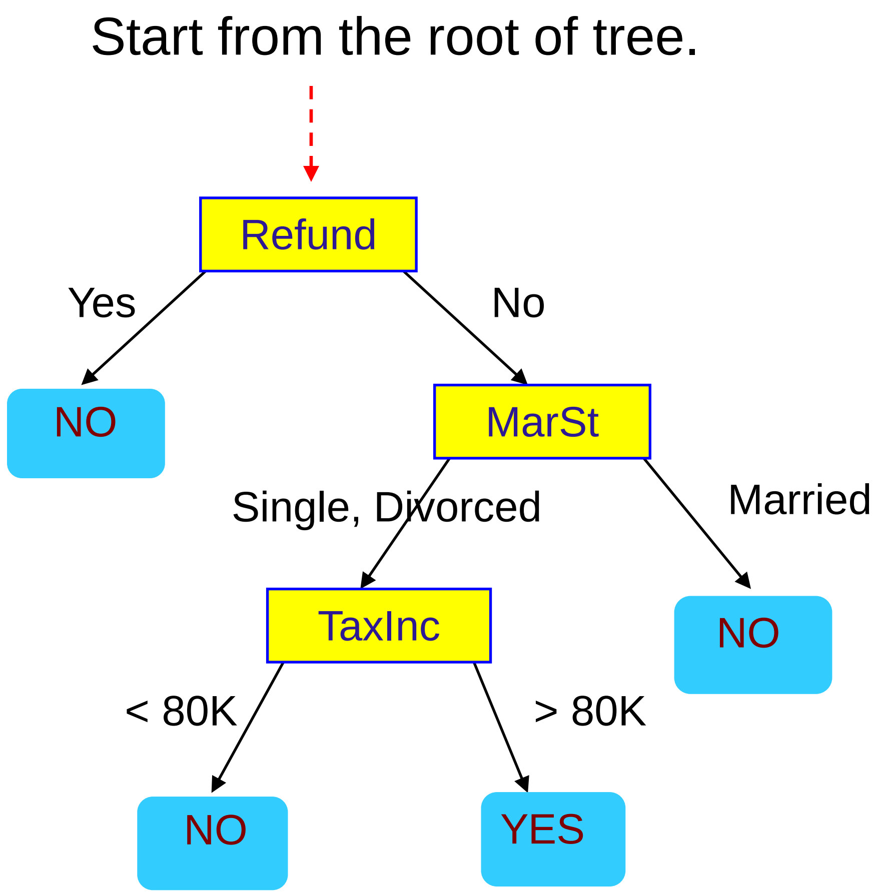
Training Set

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



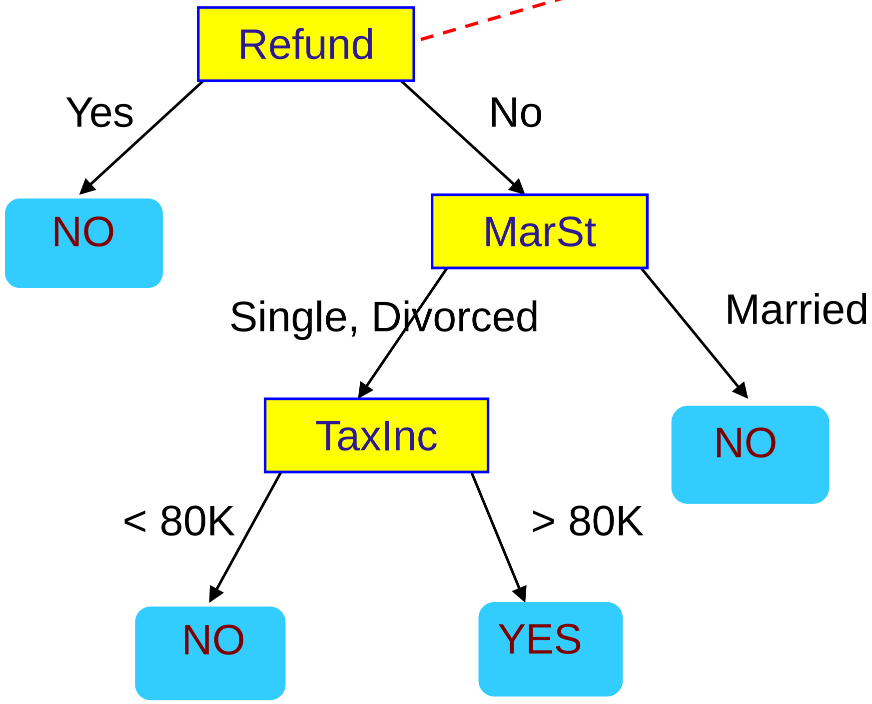
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data



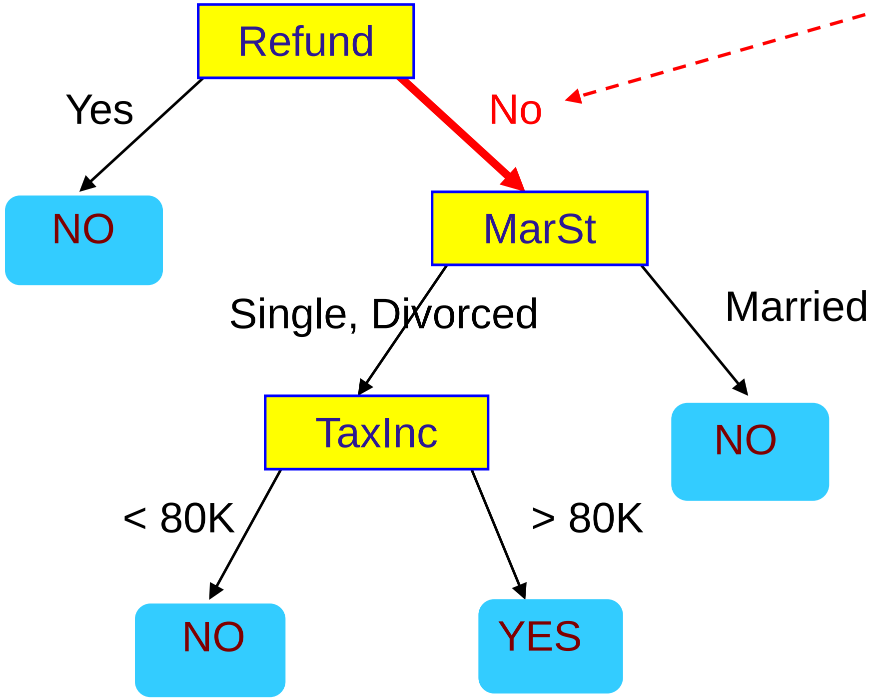
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

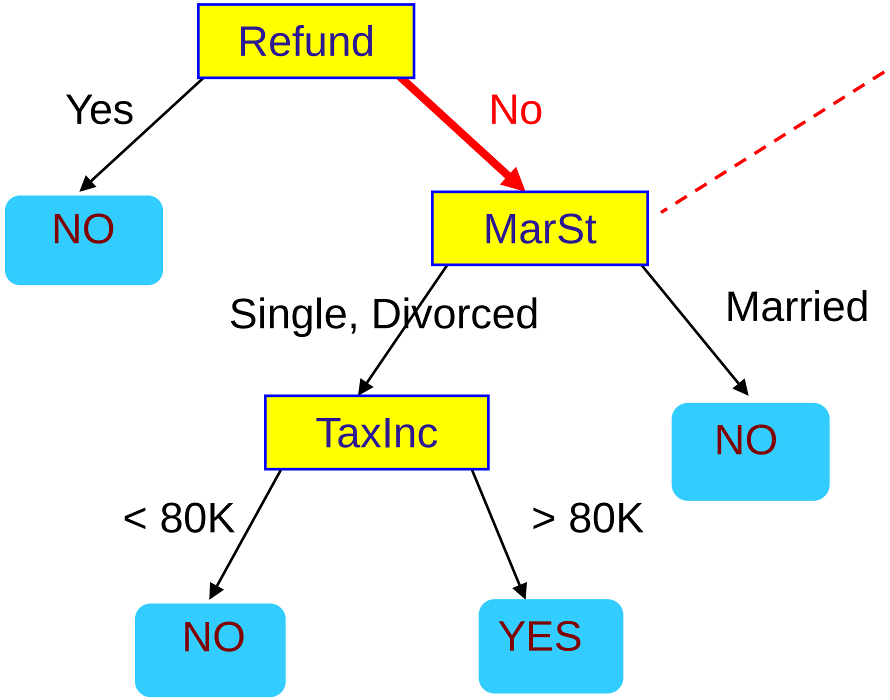
Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data



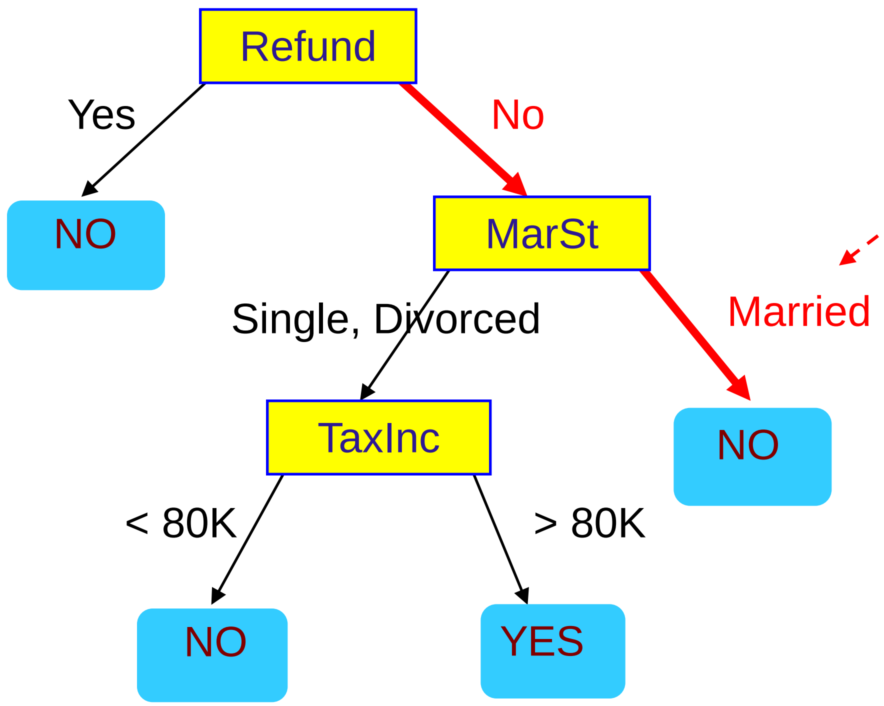
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

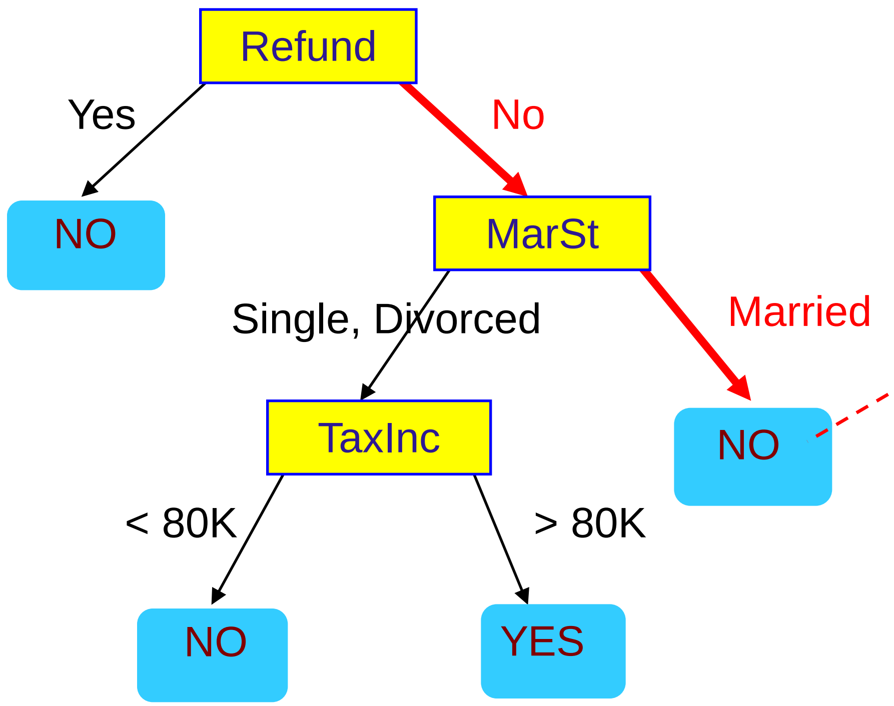
Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data



Test Data

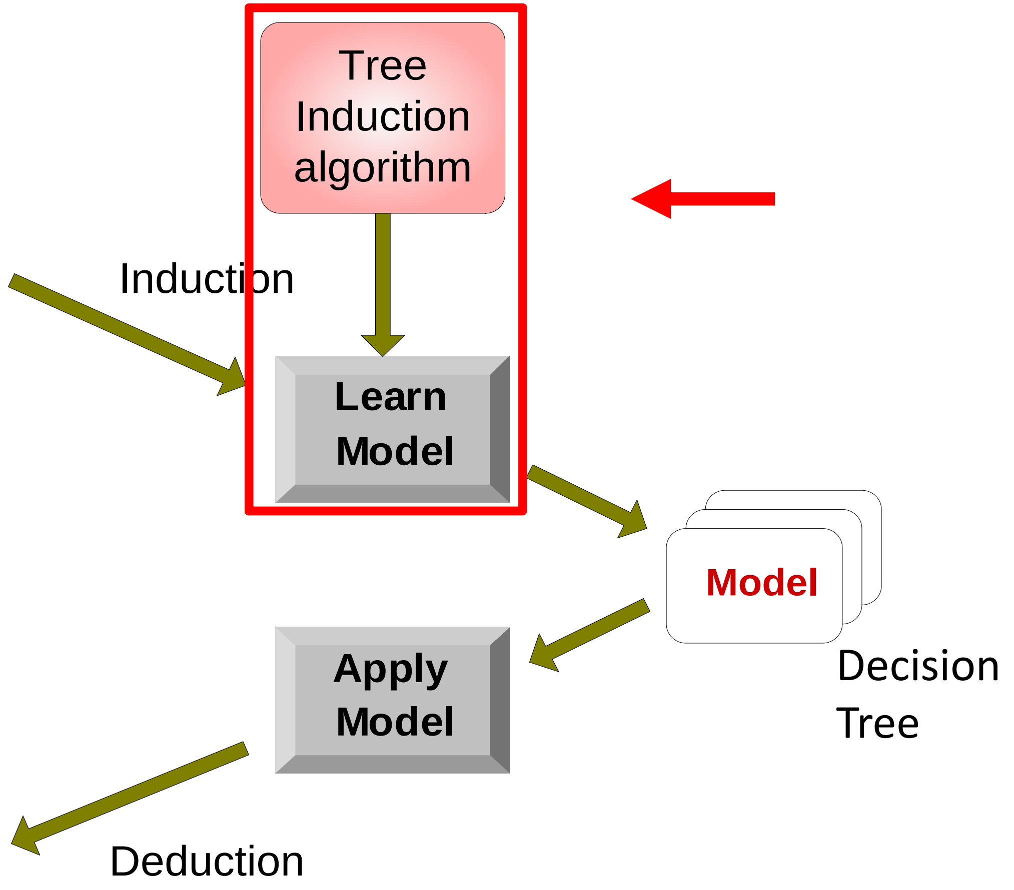
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Assign Cheat to “No”

Decision Tree Classification Task

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

Two design issues involved

- How to select the attribute(s) for the splitting ?
- When to stop the splitting in order to avoid overfitting ?

How the decision tree works

- In a decision tree, the idea is to split the data set based on **homogeneity** of data.
- The measure of impurity of a data set must be at a maximum when all possible classes are equally represented.
- The measure of impurity of a data set must be zero when only one class is represented.
- Measures such as **entropy** or **Gini index** easily meet these criteria and are used to build decision trees as described in the following sections. Different criteria will build different trees through different biases, for example, **information gain** favors tree splits that contain many cases, while **information gain ratio** attempts to balance this.

Two common decision tree algorithms

- Classification and regression tree(CART)
  - Only binary split per node
- C5.0
  - By default entropy is used for the splitting criteria
  - Could support multiple split per node

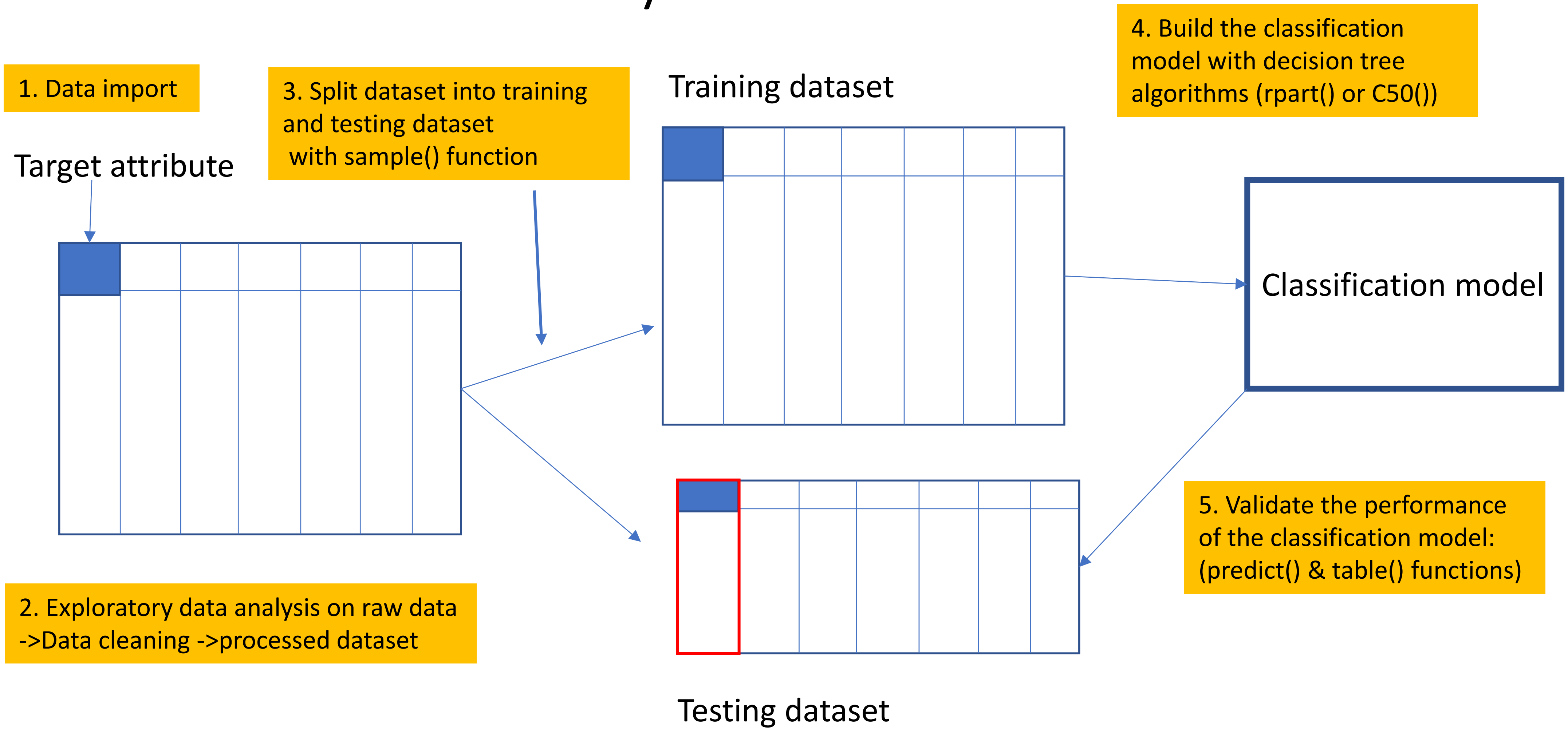
Pruning a decision tree: When to stop the splitting

- No attribute satisfies a minimum information gain threshold
- A maximal depth is reached
- Pruning is used so as to avoid overfitting
- Overfitting by a decision tree results not only in a difficult to interpret model, but also provides quite a useless model for unseen data.
- Two approaches of pruning: Pre-pruning and post-pruning

Pre-pruning and Post-pruning

- The above two stopping techniques mentioned above constitute what is known as **pre-pruning** the decision tree, because the pruning occurs before or during the growth of the tree.
- There are also methods that will not restrict the number of branches and allow the tree to grow as deep as the data will allow, and then trim or prune those branches that do not effectively change the classification error rates. This is called **post-pruning**.
- Post-pruning may sometimes be a better option because we will not miss any small but potentially significant relationships between attribute values and classes if we allow the tree to reach its maximum depth. However, one drawback with post-pruning is that it requires additional computations, which may be wasted when the tree needs to be trimmed back.

Workflow of the analysis



Use of decision tree with Scikit-learn

```
from sklearn.tree import DecisionTreeClassifier

tree = DecisionTreeClassifier(criterion='gini',
max_depth=4, random_state=1)
tree.fit(X_train, y_train)
```

Impurity measurement with Gini

- A node's gini attribute measures its impurity: a node is "pure" (gini = 0) if all training instances it applies to belong to the same class.
- For example, since the depth-1 left node applies only to Iris-Setosa training instances, it is pure and its gini score is 0.

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2$$

The CART algorithm

- Scikit-Learn uses the Classification And Regression Tree (CART) algorithm to train Decision Trees (also called "growing" trees).
- The idea is really quite simple: the algorithm first splits the training set in two subsets using a single feature k and a threshold tk (e.g., "petal length ≤ 2.45 cm").
- How does it choose k and tk? It searches for the pair (k, tk) that produces the purest subsets (weighted by their size). The cost function that the algorithm tries to minimize is given by the following equation.
- CART cost function for classification

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

$$\text{where } \begin{cases} G_{\text{left/right}} \text{ measures the impurity of the left/right subset,} \\ m_{\text{left/right}} \text{ is the number of instances in the left/right subset.} \end{cases}$$

- Once it has successfully split the training set in two, it splits the subsets using the same logic, then the sub-subsets and so on, recursively.
- It stops recursing once it reaches the maximum depth (defined by the max_depth hyperparameter), or if it cannot find a split that will reduce impurity.
- A few other hyperparameters (described in a moment) control additional stopping conditions (min_samples_split, min_samples_leaf, min_weight_fraction_leaf, and max_leaf_nodes).

Decision tree decision boundaries