CDS6214

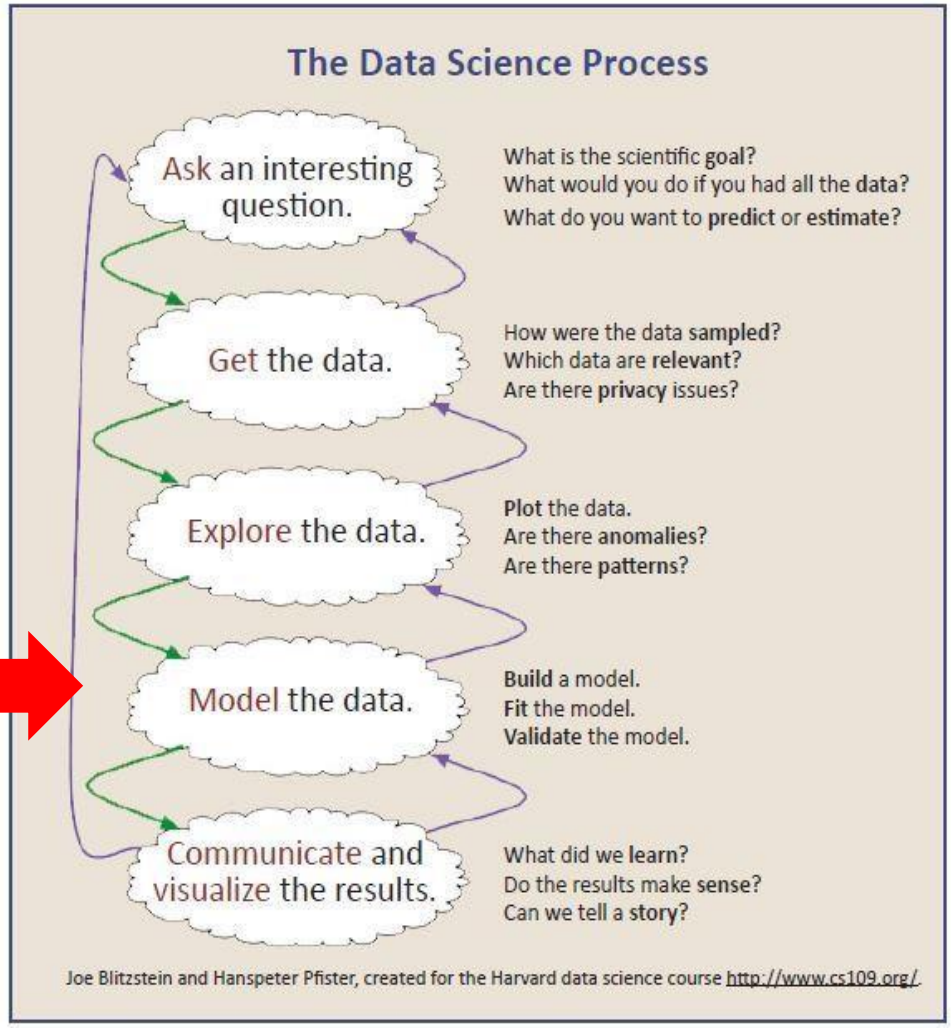
Data Science Fundamentals

Lecture 6
Predictive Modeling

Outline

- **Prediction**
  - Phases: Building & Applying
  - Classification & Regression
    - Decision Trees
    - Linear Regression
- **Classification Metrics**
- **Overfitting & Underfitting**
- **Cross-validation**

Data Science Process



What should we do next?

- Identify the question
- Collect and pre-process the data
- Explore and analyze the data
- Model the data
- Infer and visualize results

making predictions

Prediction

- Predictive models are used in many situations where an estimate or forecast is required, e.g. to project sales or forecast the weather.
  - A predictive model will calculate an estimate for one or more variables (responses), based on other variables (descriptors).
  - Example: A model to predict car fuel efficiency can be built using the MPG variable as the response and the variables Cylinders, Displacement, Horsepower, Weight, and Acceleration as descriptors.

Prediction

- A predictive model attempts to understand the relationship between the input descriptor variables and the output response variables; however, it is just a representation of the relationship.
- Rather than thinking any model generated as correct or not correct, it may be more useful to think of these models as useful or not useful to what you are trying to accomplish

Usage of Predictive Models

1. Prioritization: Predictive models can be used to swiftly profile a data set that needs to be prioritized.

- E.g. #1: A credit card company may build a predictive model to estimate which individuals would be the best candidates for a direct mailing campaign. This model could be run over a database of millions of potential customers to identify a subset of the most promising customers.
- E.g. #2: A team of scientists may be about to conduct a costly experiment and they wish to prioritize which alternative experiments have the greatest chance of success.

Usage of Predictive Models

2. Decision support: Prediction models can also be used to estimate future events so that appropriate actions can be taken.

- E.g.: Prediction models are used to forecast adverse weather conditions and that information is used to trigger events such as alerting emergency services to prepare any affected neighborhoods.

Usage of Predictive Models

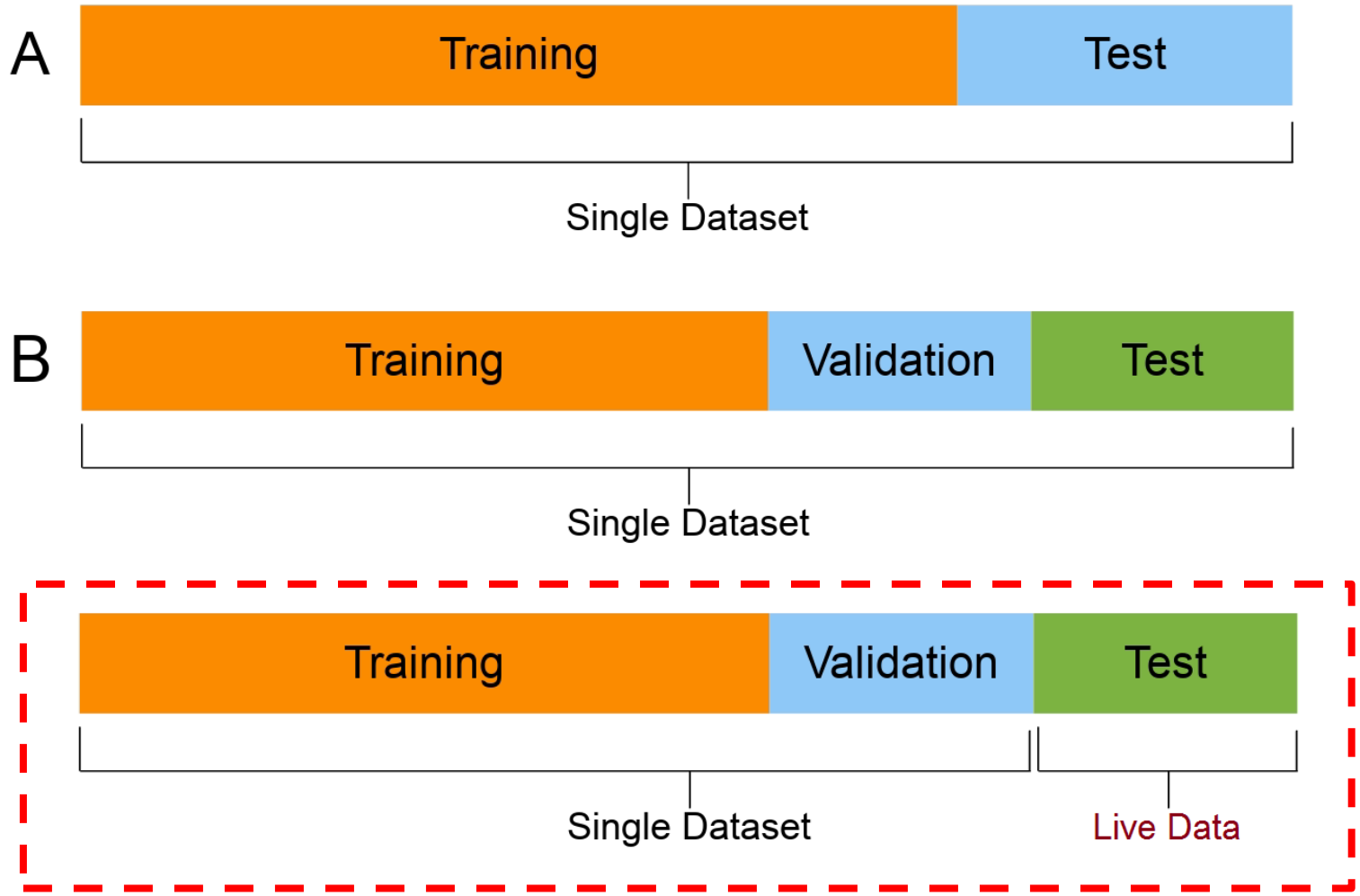
3. Understanding: Since predictive models attempt to understand the relationship between the input descriptor variables and the output response variables, they can be helpful beyond just calculating estimates

- E.g.: If a prediction model was built based on a set of scientific experiments, the model will be able to suggest what variables are most important and how they contribute to the problem under investigation.

Phases: Building & Applying

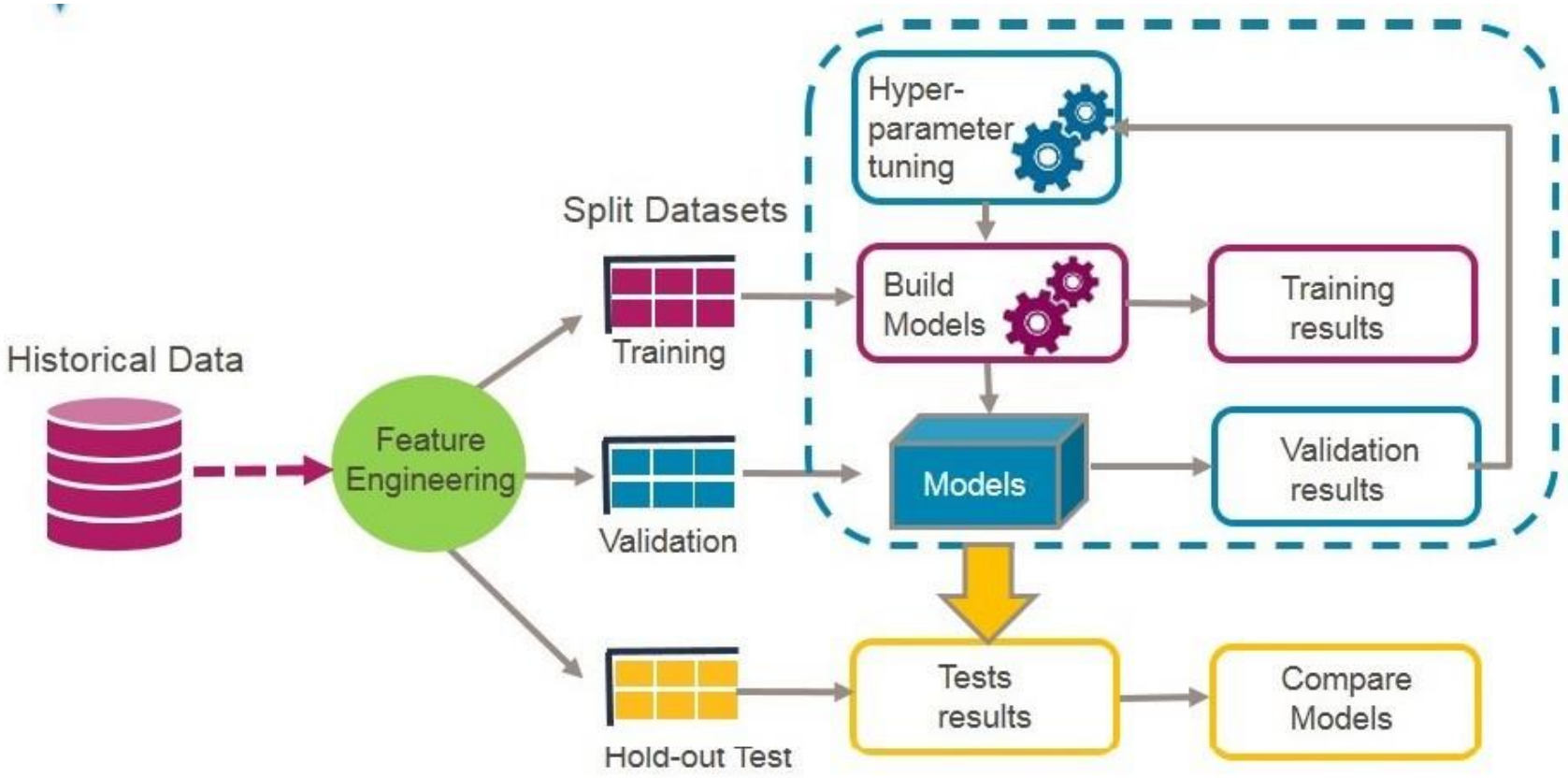
- **Building:**
  - Data used contains values for the descriptor and response variables
  - **Training set** ⇒ This set of data is used for building the model
  - **Test (or Validation) set** ⇒ This set of data is used for accessing the quality of the model. The known responses are used to gauge the prediction capability. A measure that reflects the prediction confidence is often calculated.
- **Applying:**
  - Once model has been built, a dataset with no output response variables can be fed into the model to obtain an estimated response. This is the "real" **Test set**

Dividing your data



Source: https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets

How the partitions are used



Source: https://www.linkedin.com/pulse/machine-learning-has-transformed-many-aspects-our-life-angarita

Building a Prediction Model

- Building a prediction model is an **experiment**.
  - Likely, necessary to build many models for which you do not necessary know which model will be the best later on.
  - Experiment must be appropriately designed to ensure optimal results.

Algorithms for Prediction

- There are many methods for building prediction models and they are often characterized based on the response variable.
- When the **response is a categorical variable**, the model is called a **classification model**. When **the response is a continuous variable**, then the model is a **regression model.**
- Decision trees predict a continuous response variable are called regression trees and decision trees built to predict a categorical response are called classification trees.

Classification	Regression
Classification trees	Regression trees
k-Nearest Neighbors	k-Nearest Neighbors
Logistic regression	Linear regressions
Naïve Bayes classifiers	Neural networks
Neural networks	Nonlinear regression
Rule-based classifiers	Partial least squares
Support vector machines	Support vector machines

classification & regression

Classification

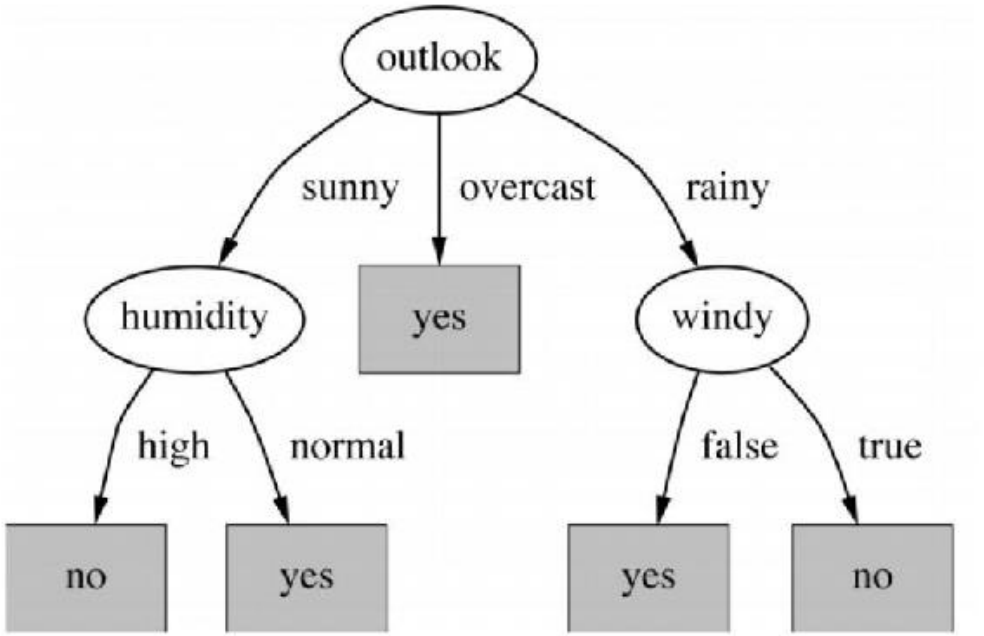
- A classification model is built to assign observations into two or more distinct categories.
- **Goal:** Assign a class to an unseen record as accurately as possible.
- Given a collection of records (training set):
  - Each record contains a set of attributes, one of the attributes is usually the "class“ or the category.
  - Find a model for class attribute as a function of the values of other attributes.

Applications of Classification

- A classification model may be built to estimate whether a customer will buy or not buy a particular product
- Predict whether drilling in a particular area will result in finding oil or not.
- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent
- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil
- Categorizing news stories as finance, weather, entertainment, sports, etc.

Decision Tree

- E.g. Descriptors: outlook, temp, humidity, windy.
  Response variable: Play Golf (Yes/No)



Decision Tree

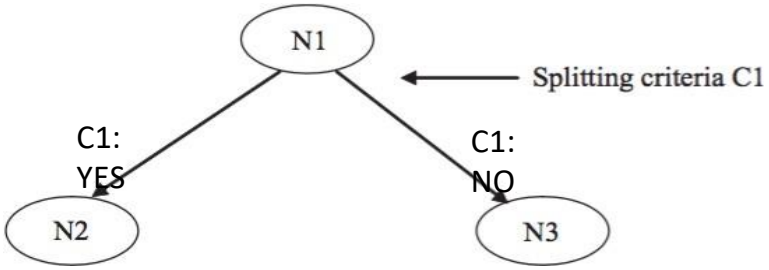
- A **tree** is made up of a series of decision points, where the entire set of observations or a subset of the observations is split based on some criteria.
  - Each point in the tree represents a set of observations called a **node**.
  - The relationship between two nodes that are joined is defined as a **parent-child** relationship.
  - The larger set which will be divided into two or more smaller sets is called the **parent node**.
  - The nodes resulting from the division of the parent are called **child nodes**.
  - A child node with no more children (no further division) is called a **leaf node.**

Decision Tree

- A table of data is used to generate a decision tree where certain variables are assigned as descriptors and one variable is assigned to be the response.
  - The descriptors will be used to build the tree, that is, these variables will divide the data set.
  - The response will be used to guide which descriptors are selected and at what value the split is made.

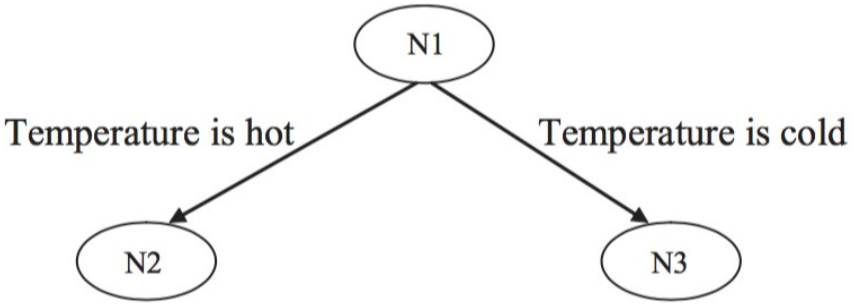
Decision Tree

- DT splits the data set into smaller and smaller sets.
  - The head (or top) of the tree is a node containing all observations.
- Analyze all descriptor variables and examine many splitting points for each variable, an initial split is made based on some criteria
- As a result, there are usually two new nodes: each node representing a smaller set of observations,

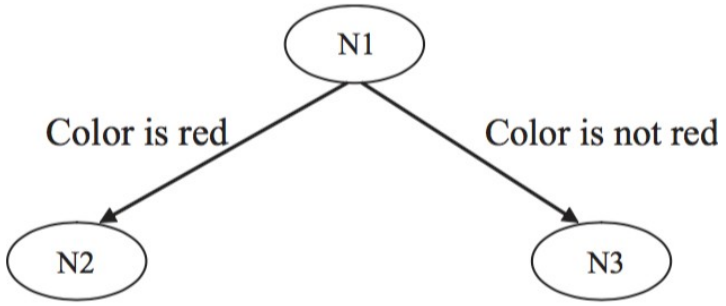


- Repeat the process of examining the variables to determine a splitting criterion for all subsequent nodes.
- A condition should be specified for ending this repetitive process. Example criteria: size of subset less than predetermined value.

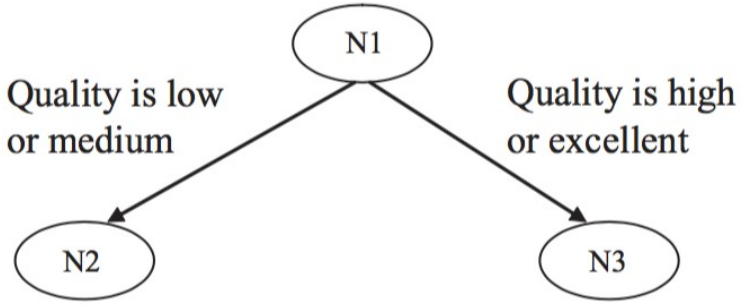
Splitting Criteria



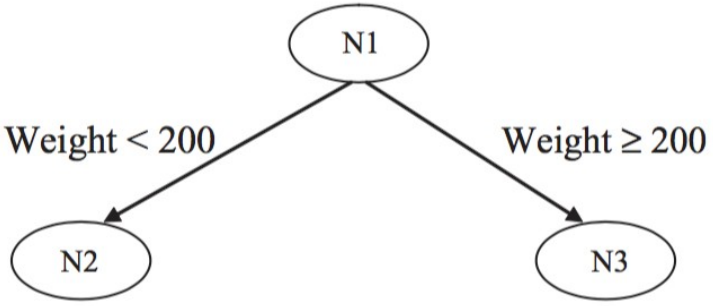
Dichotomous



Nominal

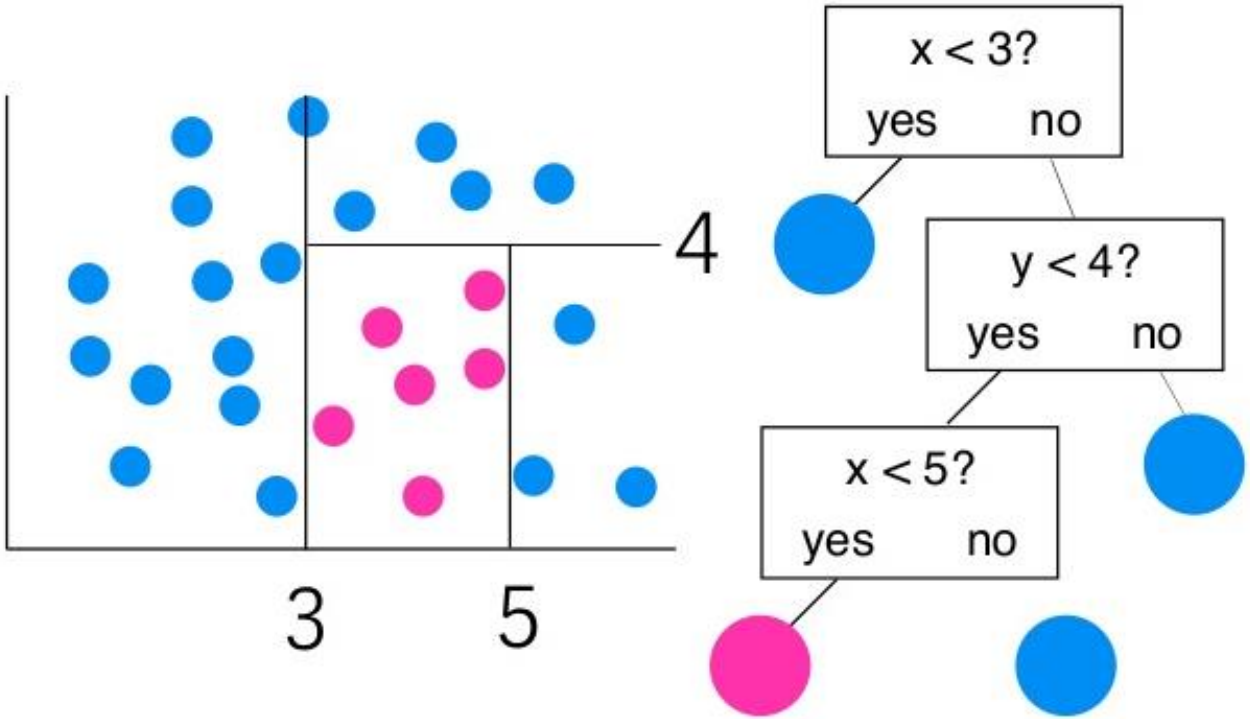


Ordinal



Continuous

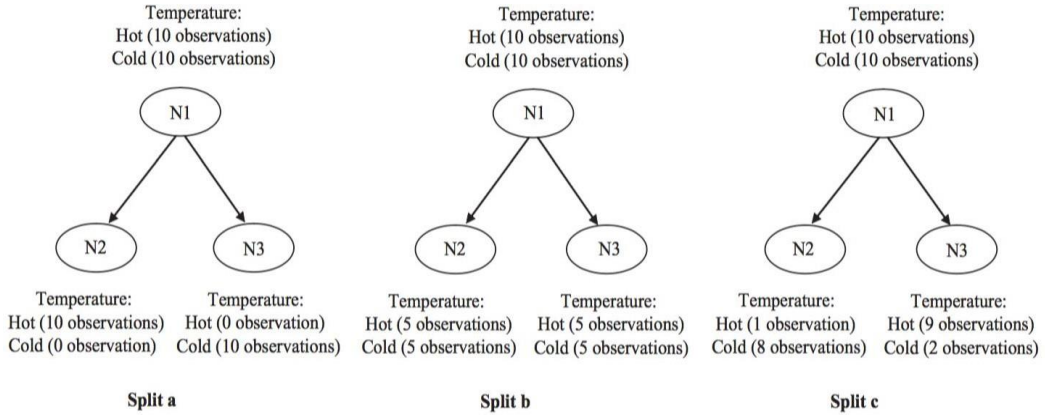
Splitting to classify data



Scoring Splits

- **Entropy:** this score reflects how well two values are split

$$Entropy(S) = -\sum_{i=1}^{c} p_i \log_2 p_i$$



- E.g. Compute the entropy values for each split

split a

$$Entropy\ (\mathrm{N1}) = -(10/20)\ \log_2\ (10/20) - (10/20)\ \log_2\ (10/20) = 1$$

$$Entropy\ (\mathrm{N2}) = -(10/10)\ \log_2\ (10/10) - (0/10)\ \log_2\ (0/10) = 0$$

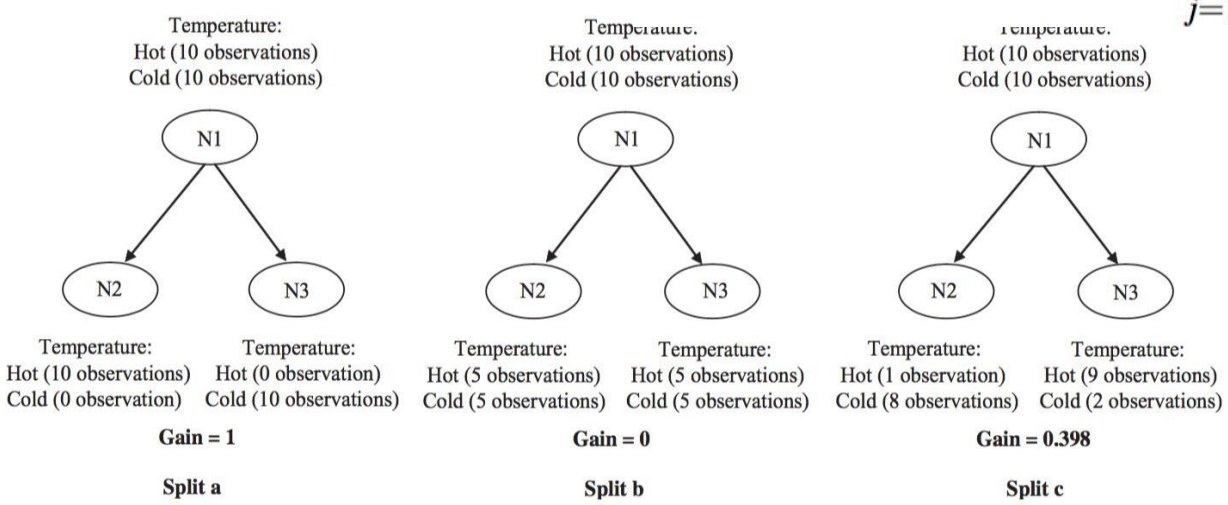
$$Entropy(\ \mathrm{N3}) = -(0/10)\ \log_2\ (0/10) - (10/10)\ \log_2\ (10/10) = 0$$

Note: There are three primary methods for calculating impurity: misclassification, Gini, and entropy.

Scoring Splits

- **Gain**: In order to determine the best split, calculate a ranking based on how cleanly each split separates the response data. This is calculated on the basis of the impurity before and after the split. As below, the criterion used in **split a** is selected as the best splitting criteria as it has the largest **gain**.

$$Gain = Entropy(parent) - \sum_{j=1}^{k} \frac{N(v_j)}{N} Entropy(v_j)$$



Scoring Splits (for Cont. Response Variables)

- **Sum of the squares of error (SSE)**: The resulting split should ideally result in sets where the response values are close to the mean of the group.
- For each potential split, a SSE value is calculated for each resulting node. A score for the split is calculated by summing the SSE values of each resulting node. Once all splits for all variables are computed, then the split with the lowest score is selected.

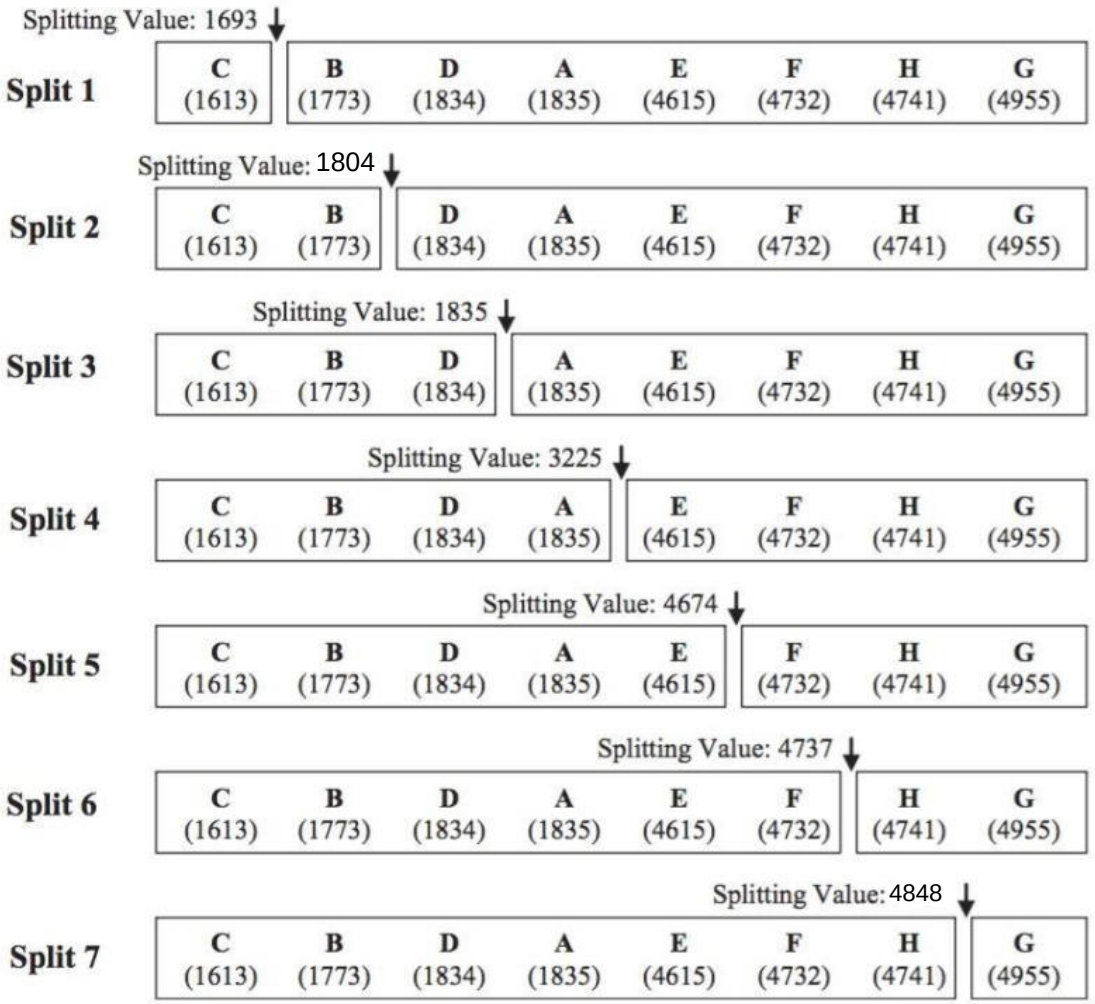
$$SSE = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Scoring Splits (for Cont. Response Variables)

	Weight	MPG
A	1,835	26
B	1,773	31
C	1,613	35
D	1,834	27
E	4,615	10
F	4,732	9
G	4,955	12
H	4,741	13

MPG is the response variable

Weight is the descriptor variable



Scoring Splits (for Cont. Response Variables)

$$SSE = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Split 3

Splitting Value: 1835

C	B	D	A	E	F	H	G
(1613)	(1773)	(1834)	(1835)	(4615)	(4732)	(4741)	(4955)

	Weight	MPG
A	1,835	26
B	1,773	31
C	1,613	35
D	1,834	27
E	4,615	10
F	4,732	9
G	4,955	12
H	4,741	13

- Example:

Split 3:

For subset where **Weight** is less than 1835 (C, B, D):

$$\text{Average} = (35 + 31 + 27)/3 = 31$$

$$\text{SSE} = (35 - 31)^2 + (31 - 31)^2 + (27 - 31)^2 = 32$$

For subset where **Weight** is greater than or equal to 1835 (A, E, F, H, G):

$$\text{Average} = (26 + 10 + 9 + 13 + 12)/5 = 14$$

$$\text{SSE} = (26-14)^2 + (10-14)^2 + (9-14)^2 + (13-14)^2 + (12-14)^2 = 190$$

$$\text{Split score} = 32 + 190 = 222$$

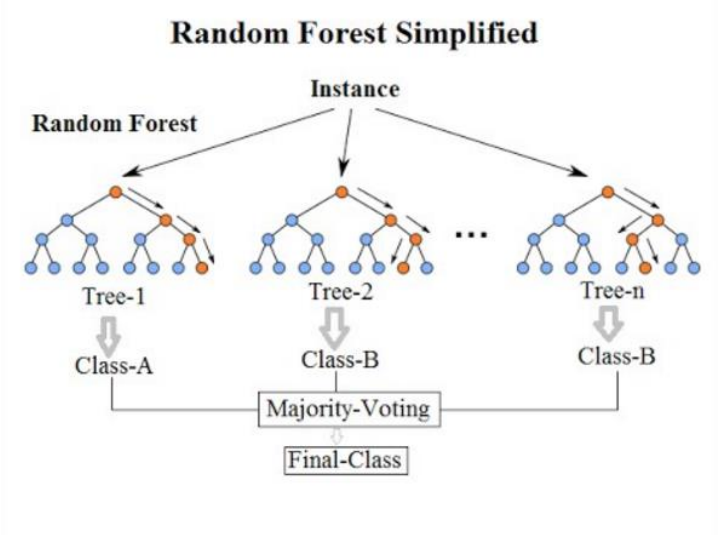
In this example, Split 4 (in previous slide) has the lowest SSE score and would be selected as the best split.

Decision Trees: Pros & Cons

- There are many *reasons to use decision trees*:
  - **Easy to understand**: Decision trees are widely used to explain how decisions are reached based on multiple criteria.
  - **Categorical and continuous variables**: Decision trees can be generated using either categorical data or continuous data.
  - **Complex relationships**: A decision tree can partition a data set into distinct regions based on ranges or specific values.
- The *disadvantages of decision trees* are:
  - **Computationally expensive:** Building decision trees can be computationally expensive when analyzing a large data set with many continuous variables.
  - **Difficult to optimize:** Generating a useful decision tree automatically can be challenging, since large and complex trees can be easily generated. Trees that are too small may not capture enough information. Generating the 'best' tree through optimization is difficult.

Random Forest

- Samples many decision trees from various subsets of the data ("bagging") and uses them as an "ensemble" of predictors. Often, some additional step is required to aggregate the predictions coming from all predictors.
- Normally performs better than a single decision tree, but at expense of more computations



Source: https://www.r-bloggers.com/machine-learning-basics-random-forest/

Other Classification Algorithms

- Logistic Regression (yes, it's not for regression!)
- Naive Bayes Classifier
- Support Vector Machines (SVM)
- Neural Networks

Source: https://www.r-bloggers.com/machine-learning-basics-random-forest/

Linear Regression

- Simple linear regression is a statistical method that allows us to summarize and study relationships between two continuous (quantitative) variables:
  - One variable, denoted **x**, is regarded as the predictor, explanatory, or independent variable.
  - The other variable, denoted **y**, is regarded as the response, outcome, or dependent variable.
- Where there appears to be a linear relationship between two variables, a simple linear regression model can be generated.

Applications of Regression

- Predict the size of mammals based on their metabolic rate.
- Predict the prices of house given the location of house, size of the house and other kinds of data
- Predict the retail sales of a particular month, given some economic indicators
- Predict the number of passengers that ride the main bus line given the weather information

Simple Linear Regression

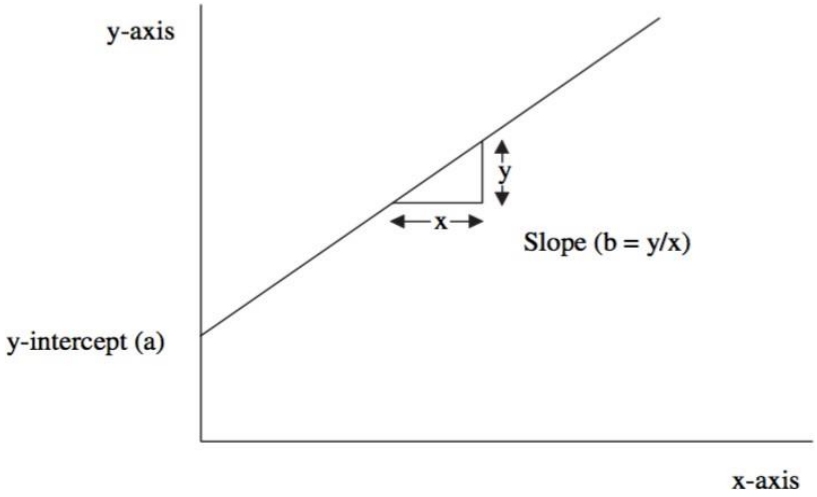
$$\hat{Y} = a + bx$$

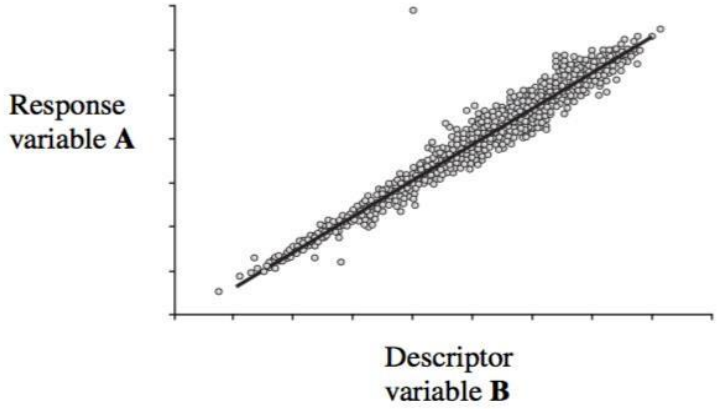
Y = dependent variable

X = independent variable

a = Y-intercept

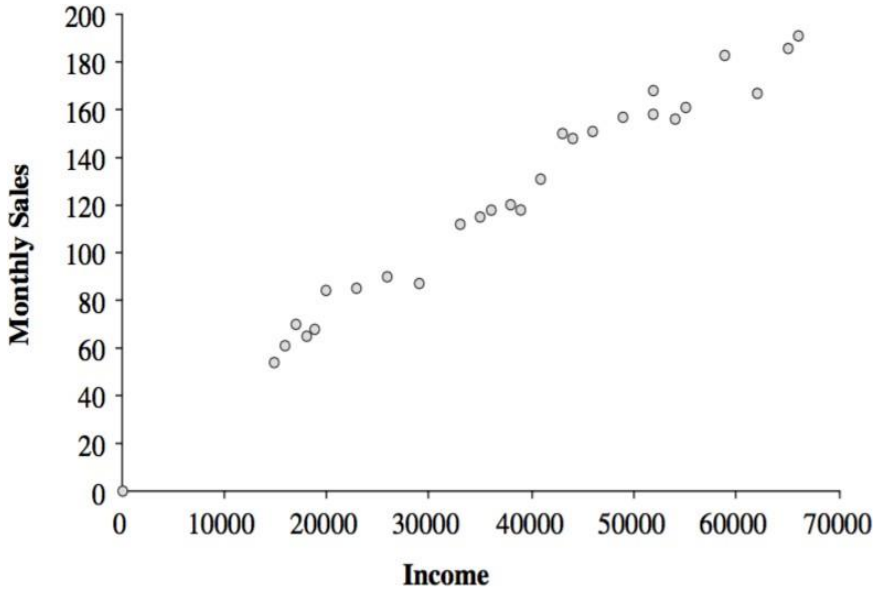
b = slope of the line





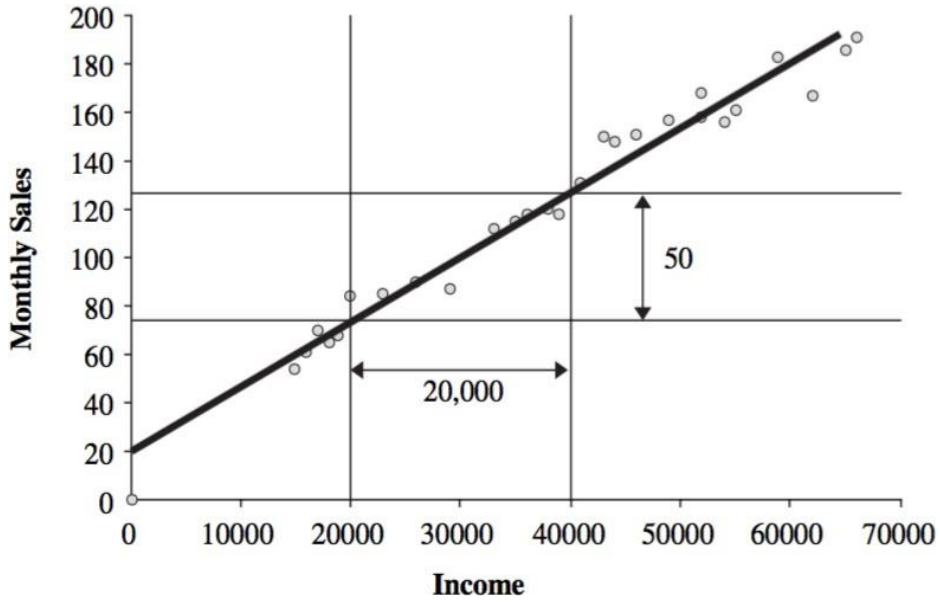
Simple Linear Regression

Income (x)	Monthly Sales (y)
$15,000.00	$54.00
$16,000.00	$61.00
$17,000.00	$70.00
$18,000.00	$65.00
$19,000.00	$68.00
$20,000.00	$84.00
$23,000.00	$85.00
$26,000.00	$90.00
$29,000.00	$87.00
$33,000.00	$112.00
$35,000.00	$115.00
$36,000.00	$118.00
$38,000.00	$120.00
$39,000.00	$118.00
$41,000.00	$131.00
$43,000.00	$150.00
$44,000.00	$148.00
$46,000.00	$151.00
$49,000.00	$157.00
$52,000.00	$168.00
$54,000.00	$156.00
$52,000.00	$158.00
$55,000.00	$161.00
$59,000.00	$183.00
$62,000.00	$167.00
$65,000.00	$186.00
$66,000.00	$191.00



Simple Linear Regression

- To manually generate a linear regression formula, a straight line is drawn through the points .



Monthly sales
= 20 + 0.0025 x **Income**

- Once a formula for the straight line has been established, predicting values for the *y* response variable based on the *x* descriptor variable can be easily calculated.

Simple Linear Regression

- Parameters *a* and *b* can be derived manually by drawing a best guess line through points in the scatterplot and then visually inspecting where the line crosses the y-axis (a) and measuring the slope (b).
- The **least squares method** is able to calculate these parameters automatically (determine the equation for the line of best fit)

- Slope

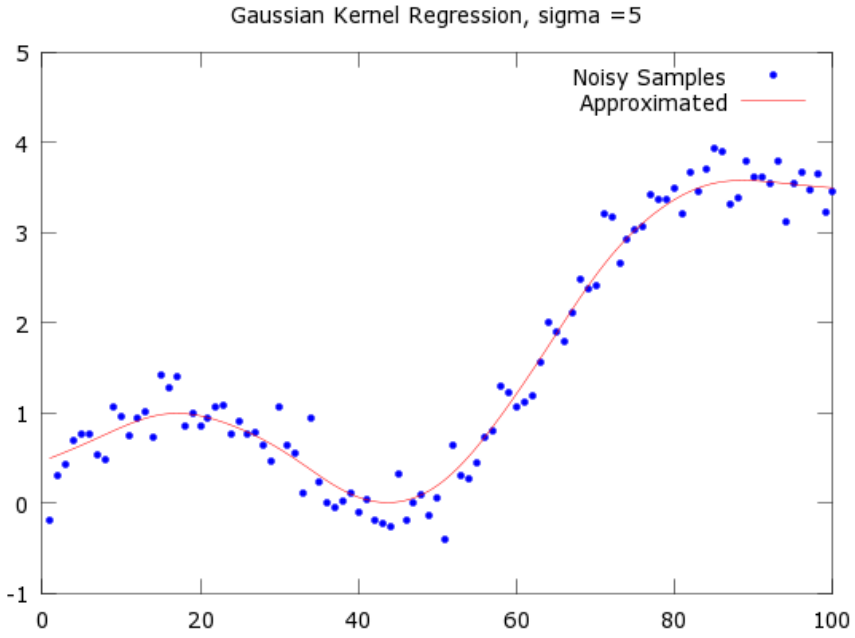
$$b = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

Y-Intercept

$$a = \bar{y} - b\bar{x}$$

Non-Linear Regression

- If there is no linear relationship between these two variables, hence a linear model may not be suitable.
- We can transform x or y or both to create a linear relationship. This is sometimes called "kernel regression" (values are passed thru a "kernel")



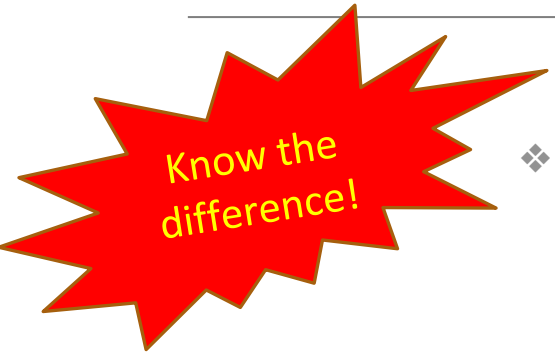
Multiple linear regression

- Multiple linear regression attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable x is associated with a value of the dependent variable y.

$$Y = a + b_1x_1 + b_2x_2 + b_3x_3 \ldots + b_nx_n$$

More often than not, we still call this **linear regression**, just that the variable x can take on multiple dimensions.

Correlation Analysis vs Linear regression



- Correlation quantifies the degree to which two variables are related. Correlation does not fit a line through the data points. We compute a correlation coefficient (r) that tells us how much one variable tends to change when the other one does.
- Linear regression finds (and describes) the best line that predicts Y from X. Correlation does not fit a line.

prediction metrics

Evaluation: Classification Metrics

- **Classification Accuracy** is the number of correct predictions made as a ratio of all predictions made.
  - Most common evaluation metric for classification problems · also the most misused!
  - It is really only suitable when there are an equal number of observations in each class (which is rarely the case) and that all predictions and prediction errors are equally important, which is often not the real-world case.
- **Logarithmic Loss** measures the predictions of probabilities of membership to a given class. The scalar probability between 0 and 1 can be seen as a measure of confidence for a prediction by an algorithm. Predictions that are correct or incorrect are rewarded or punished proportionally to the confidence of the prediction. Often used with logistic regression.

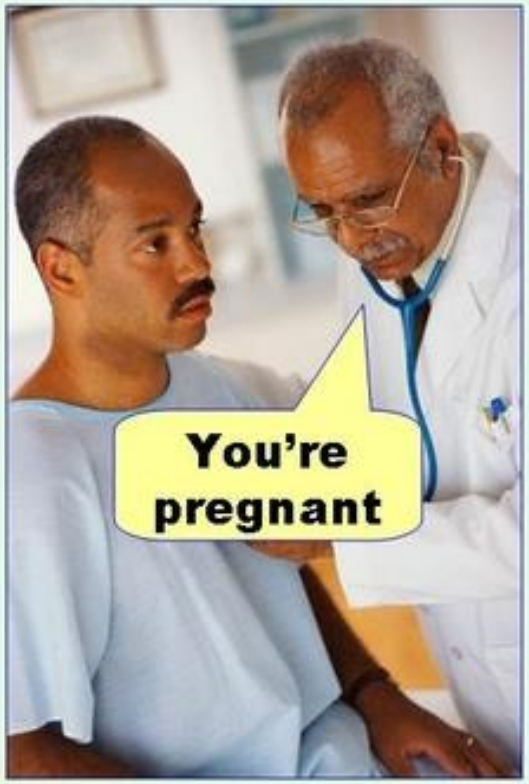
Evaluation: Classification Metrics

- **Confusion Matrix** is a handy presentation of the performance of a model with two or more classes. The table presents the number of predictions made by the algorithm and number of actual class samples on two opposing sides, for all classes.
  - True positive (TP) = correctly identified
  - False positive (FP) = incorrectly identified
  - True negative (TN) = correctly rejected
  - False negative (FN) = incorrectly rejected

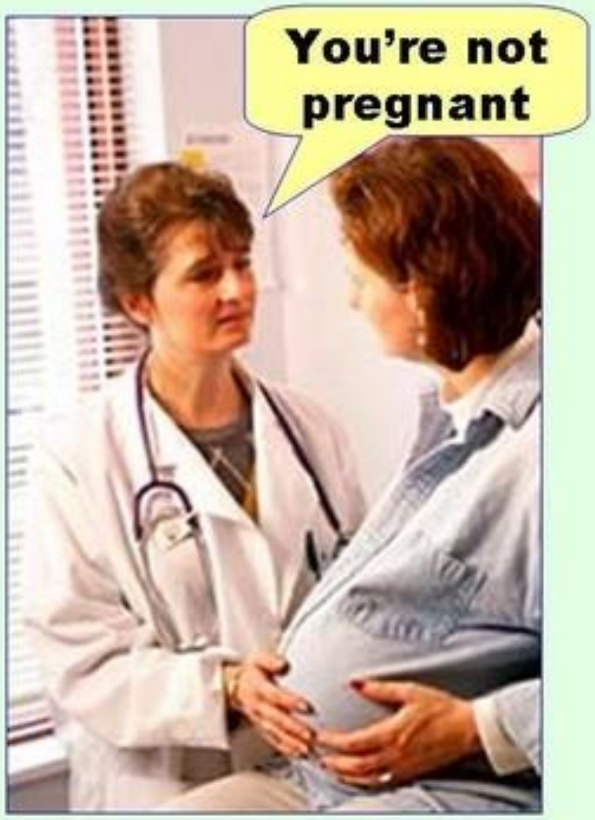
n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

False Positive vs. False Negative

(false positive)



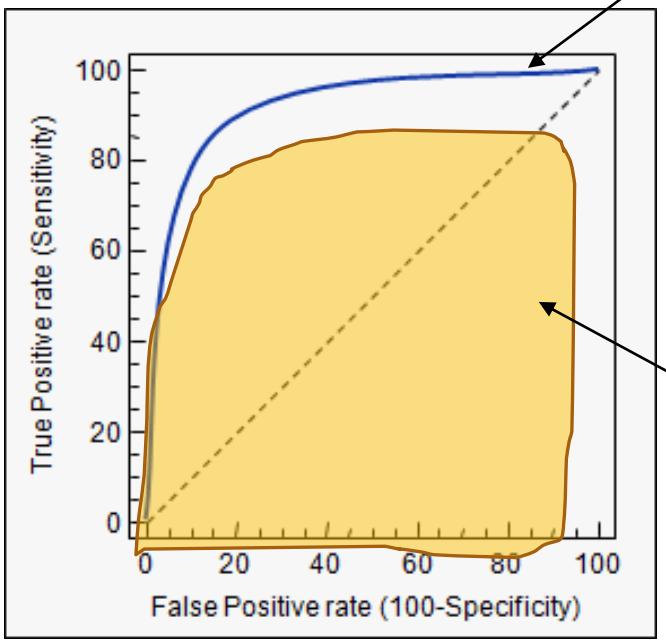
(false negative)



n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Evaluation: Classification Metrics

Receiver Operating Characteristic (ROC) Curve : TPR vs FPR plot



$$\text{True Positive Rate} = \frac{TP}{TP+FN}$$

$$\text{False Positive Rate} = \frac{FP}{FP+TN}$$

Area Under Curve (AUC)

Evaluation: Classification Metrics

- **Area Under ROC Curve** (or AUC for short) is a performance metric for binary classification problems. The AUC represents a model's ability to discriminate between positive and negative classes. An area of 1.0 represents a model that made all predictions perfectly. An area of 0.5 represents a model as good as random.

Evaluation: Classification Metrics

- **Precision**
- **Recall**
- **F1-score** (in between the precision and recall scores)
  - Suitable for imbalanced datasets

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

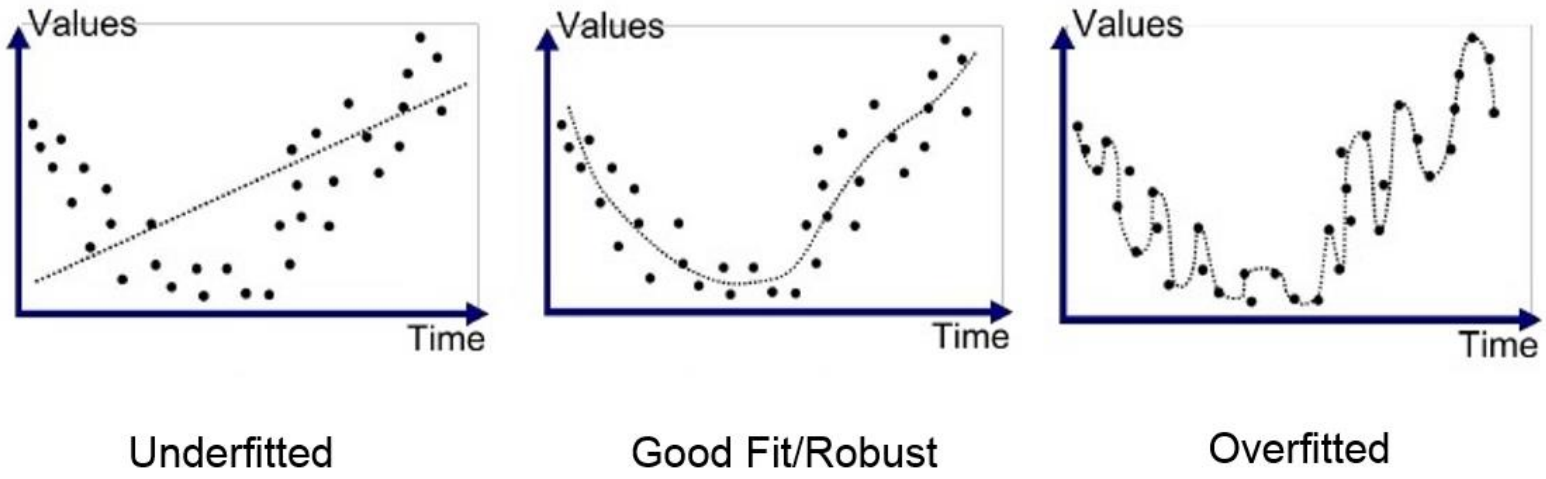
$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

Evaluation: Regression Metrics

- **The Mean Absolute Error (or MAE)** is the sum of the absolute differences between predictions and actual values. It gives an idea of how wrong the predictions were.
  - Indicates only magnitude of the error, but no idea of the direction of error (e.g. over or under predicting).
- **The Mean Squared Error (or MSE)** is provides a general idea of the magnitude of error along all dimensions.
- **R^2 (or R Squared)** metric provides an indication of the goodness of fit of a set of predictions to the actual values. This is a value between 0 and 1 for no-fit and perfect fit respectively.

diagnostics for modelling

Over-fitting & Under-fitting

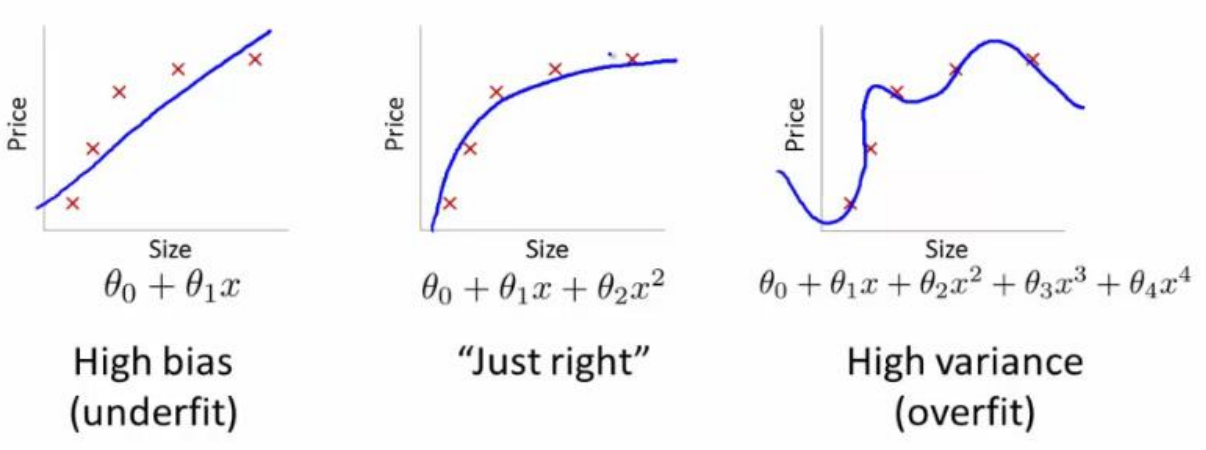


- A model that is too simple ⇒ Under-fits data
  - Cannot attain sufficient accuracy, caters too generally
- A model that is too complex ⇒ Over-fits data
  - Too accurately fitting to the training data, may not generalize well with new data

Source: https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76

Over-fitting & Under-fitting

- From the perspective of a regression problem:



- Using more variables lead to higher complexity in the model
- Remedies:
  - Underfitting: features/params ↑
  - Over-fitting: features/params ↓ or data ↑

Over-Fitting

- Over-fitting refers to a model that models the training data too well.
  - A model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. Noise or random fluctuations in the training data is picked up and learned by the model. This negatively impact the model's ability to generalize.
- E.g. Decision trees algorithm is very flexible and is subject to over-fitting training data. This problem can be addressed by pruning a tree after it has been learned in order to remove some of the fine divisions it has picked up.
- Two important techniques that you can use when evaluating prediction algorithms to limit overfitting:
  - Use a resampling technique (cross-validation) to estimate model accuracy.
  - Hold back a validation dataset to tune model parameters

Cross-Validation

- In k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, while the remaining k – 1 subsamples are used as training data.
- The cross-validation process is then repeated k times (i.e. folds), with each of the k subsamples used exactly once as the validation data.
- The k results from the folds can then be averaged to produce a single estimation of the model's performance.

Cross-Validation

Feature Engineering

- **Dimensionality reduction ⇒** typically choosing a basis or mathematical representation within which you can describe most but not all of the variance within your data, thereby retaining the relevant information, while reducing the amount of information necessary to represent it.
  - A variety of techniques: PCA, ICA, Matrix Feature Factorization, etc.
  - Existing data is reduce to its most discriminative information. This allows us to represent most of the information in the dataset with fewer, but more discriminative features.
- **Feature Selection:** Selecting features which are highly discriminative. It requires an understanding of what aspects of your dataset are important in whatever predictions you're making, and which are not. Correlation analysis can help in feature selection.
- **Feature extraction:** Generating or extracting new features which are composites of the existing features, which may be better and more discriminative

Additional Resources

- Scikit-learn tutorial, "Supervised Learning". Very comprehensive and can be a whole course of its own!
- Jason Brownlee, http://machinelearningmastery.com
- Book code repositories on GitHub:
  - Sebastian Raschka's "Python Machine Learning"
  - Andreas Mueller's "Introduction to Machine Learning with Python"