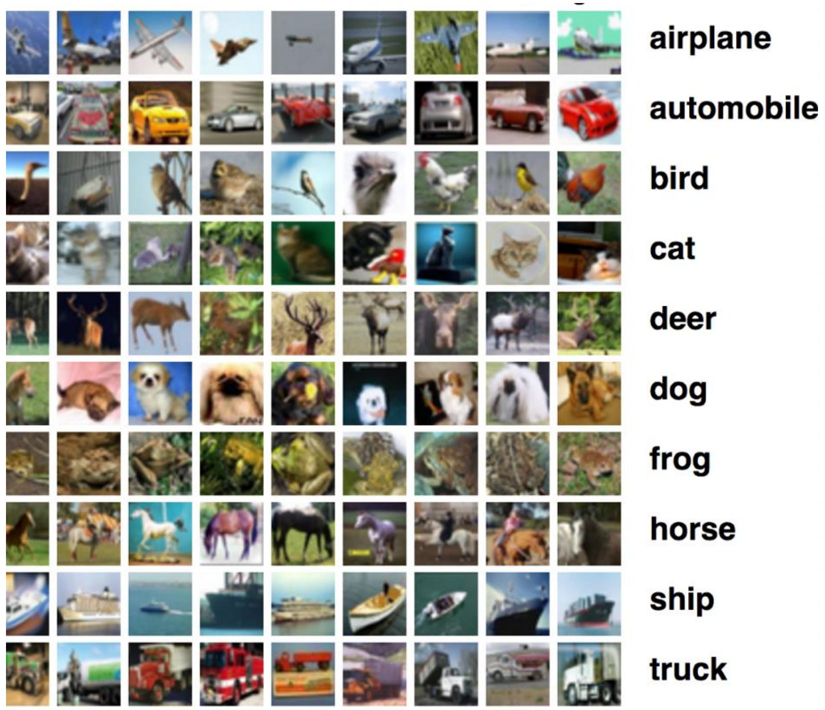
ECE 4252/8803: Fundamentals of Machine Learning (FunML)
Fall 2024

Lecture 2: Classification



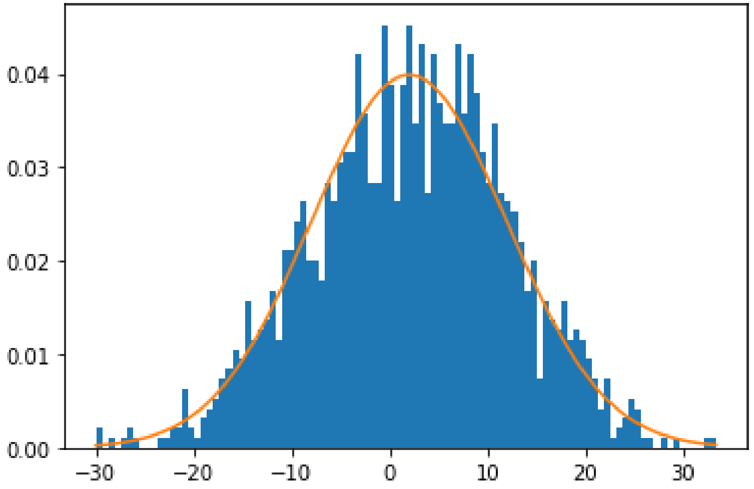
Outline

Agenda: Intro to Classification

- Lecture 1
  - Canvas and Logistics
  - Learning Algorithm
  - Task
  - Performance
  - Experience
- Classification Overview
  - The Classification Challenges
  - Predictor Classifier Modeling
  - Types of Classification Models
  - Terminologies
- Nearest Neighbor Classifier
- Naïve Bayes Classifier

Unsupervised learning algorithms

- Dataset contains many features
- learn useful properties of the structure of this dataset such as probability distribution or perform other roles, like clustering, which consists of dividing the dataset into clusters of similar examples.
- for instance, by observing several instaces of a random variable, $\mathbf{x}$, we might be able to estimate $p(\mathbf{x})$



Material adapted from www.deeplearningbook.org

Supervised learning algorithms

- Dataset contains features, but each example is also associated with a label or target.
- For example, Iris dataset is annotated with the species of each iris plant. A supervised learning algorithm can study the Iris dataset and learn to classify iris plants into three different species based on their measurements
- Given a few instances of $\mathbf{x}$, and their correct "labels", $\mathbf{y}$, we want to learn $p(\mathbf{y}|\mathbf{x})$
- The term supervised learning originates from the view of the label being provided by an instructor or teacher who shows the machine learning system what to do

Classification

Overview

- The Classification Problem
- Predictive Classifier Modeling
- Types of Classification Models
- Terminology

Overview

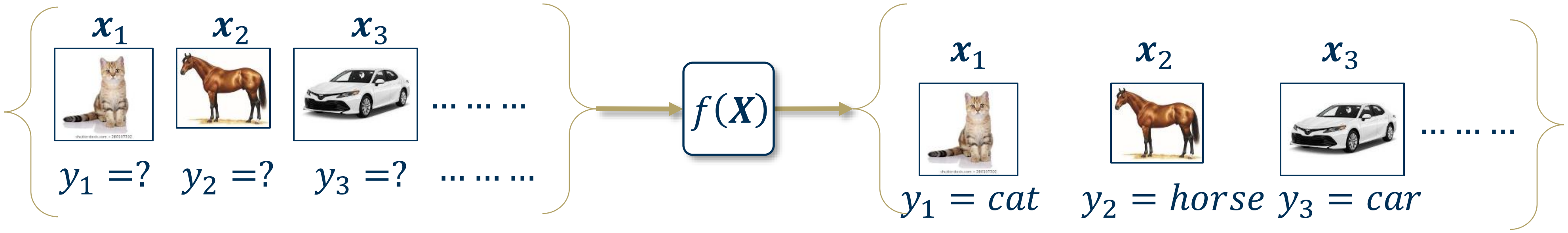
What is Classification?

Finds decision boundary(ies) between labeled data and uses that boundary to label new objects.

- The task of approximating a mapping function f from input variables $\boldsymbol{X}$ to discrete output variables $\boldsymbol{y}$.
- That is, given a set of class labels $\{\boldsymbol{y}\}$, and a set of objects $\{\boldsymbol{X}\}$, the goal of classification is to labels objects in $\{\boldsymbol{X}\}$ with their correct class in $\{\boldsymbol{y}\}$

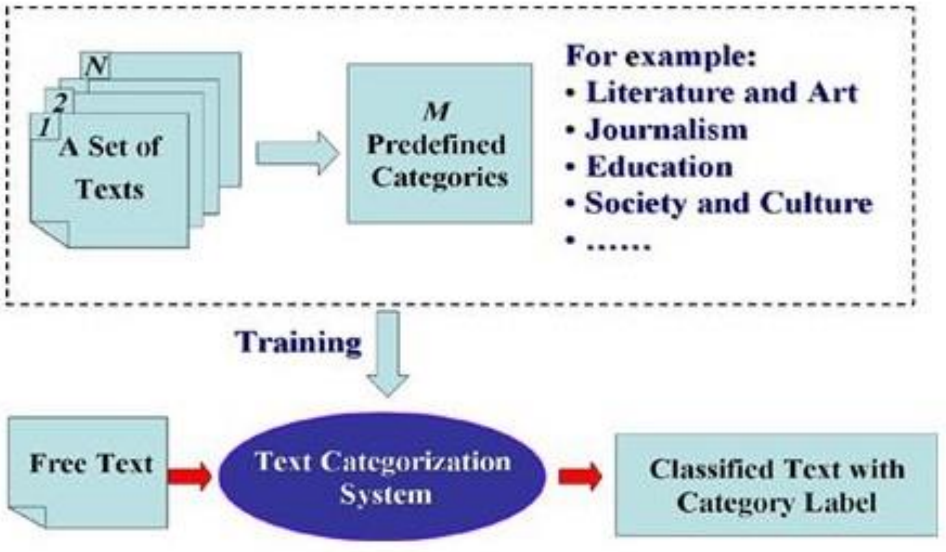
$\{y\}$

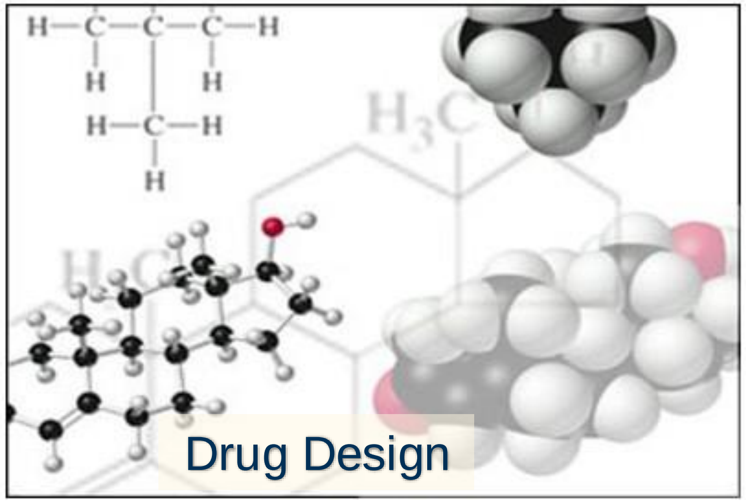
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

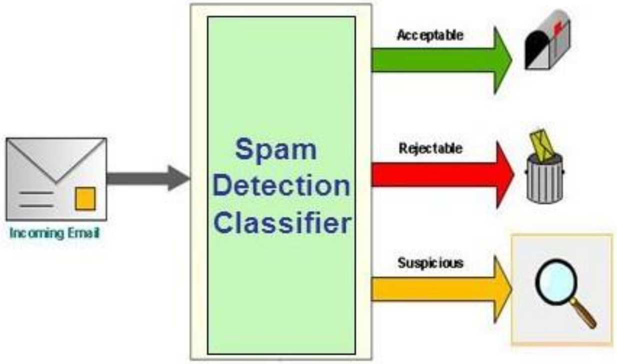


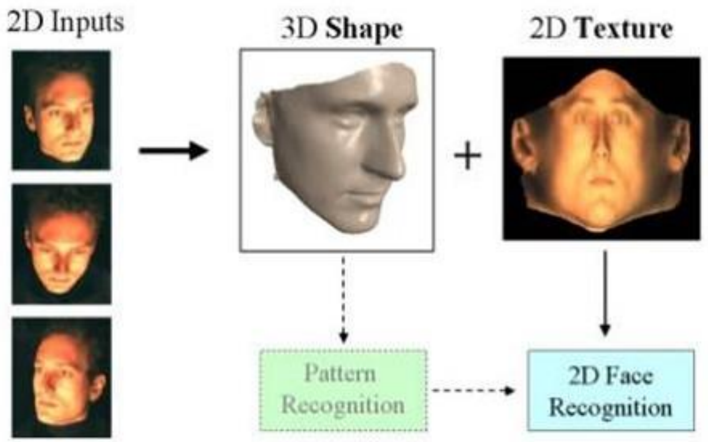
Overview

Examples of Classification







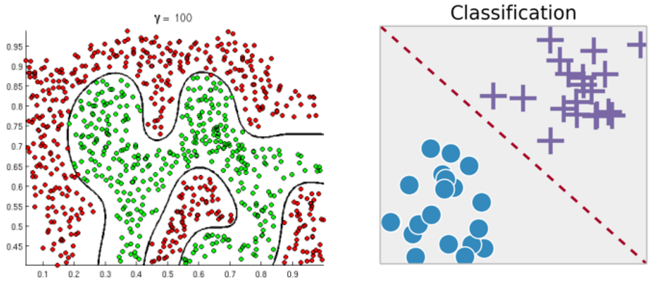


Overview

Classification vs Regression vs Clustering

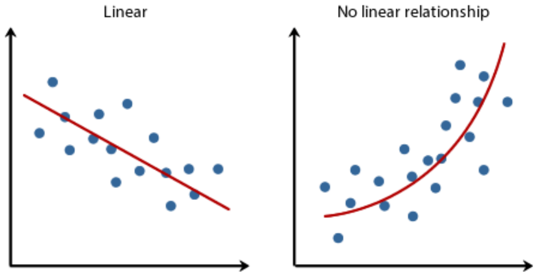
- **Classification:**

  Finds decision boundary(ies) between labeled data and uses that boundary to label new objects.



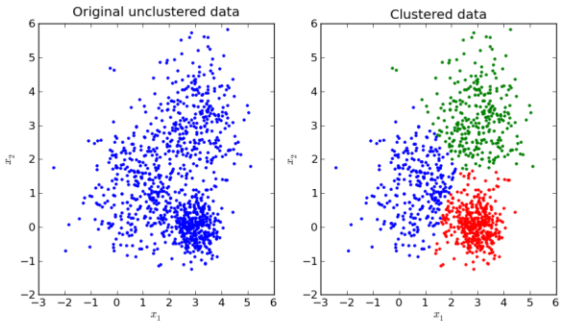
- **Regression:**

  Finds a linear or non-linear real-valued function to predict mapping to a single value



- **Clustering:**

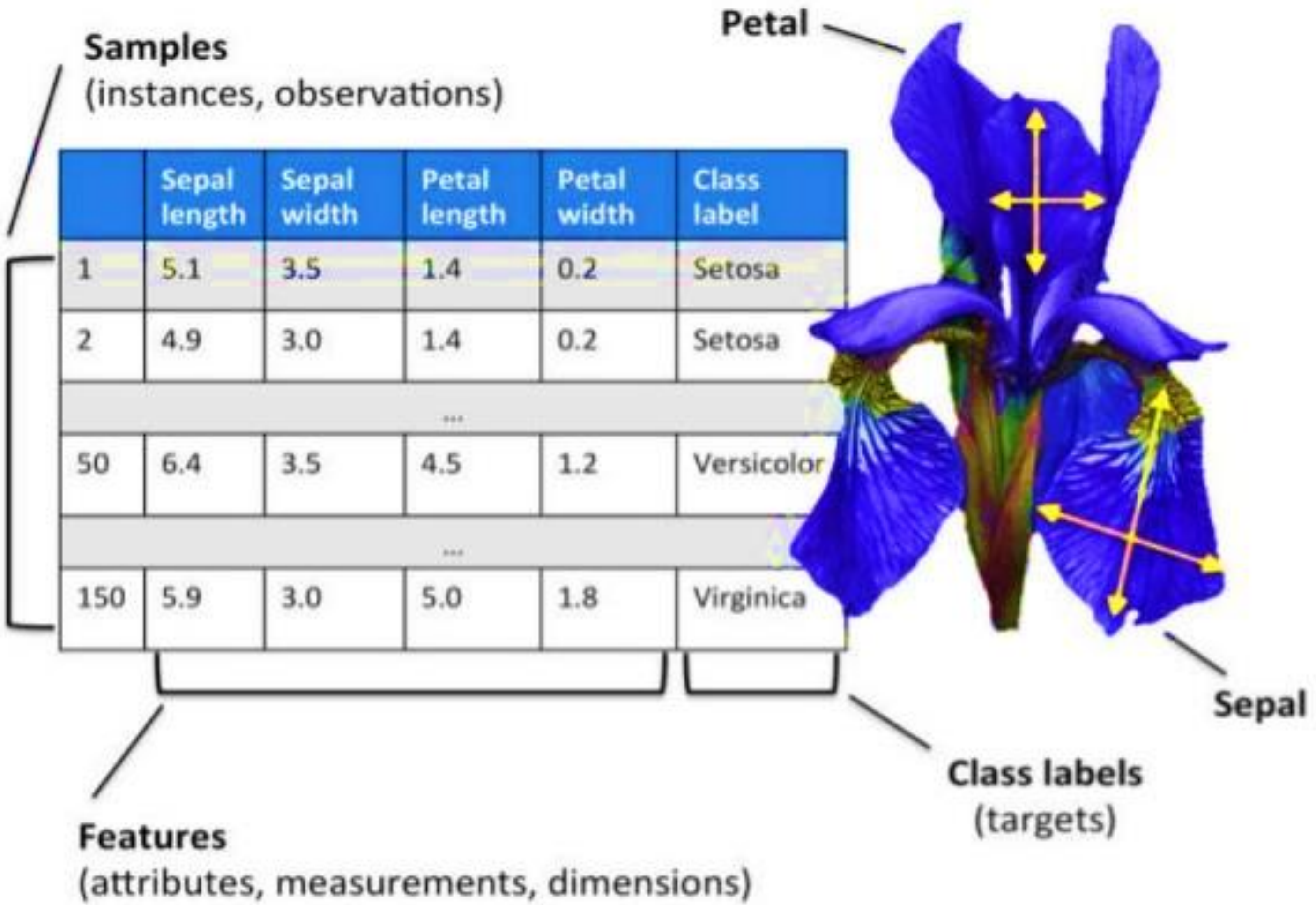
  Grouping together similar non-labeled data into a number of clusters. The number of clusters is generally pre-determined, but it could be estimated by the clustering method.



Overview

Terminologies on IRIS Dataset

- Samples
- Features
- Feature Vector
- Feature Extraction
- Target class
- Learning Model
- Training Algorithm
- Training/Evaluation set



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

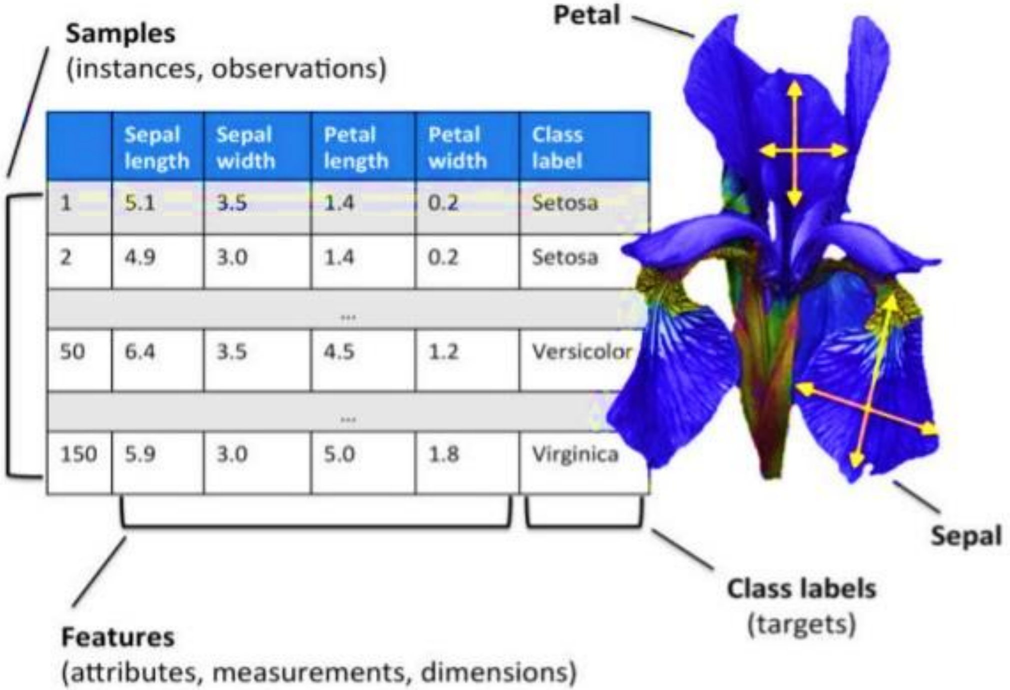
Overview

Terminologies on IRIS Dataset

- Samples: A sample is a data item to process (classify). It can be a document, a picture, audio clip, video, row in database or CSV file, or whatever you can describe with a fixed set of quantitative traits.

- Features: The collection of distinct traits that can be used to describe each data item in a quantitative manner

- Feature Vector: An *n*-dimensional vector that contains the concatenation of all features that represent a sample



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

Overview

Terminologies

- Feature Extraction: A step that transforms the raw data in the high-dimensional space to a space of fewer dimensions (feature vector)

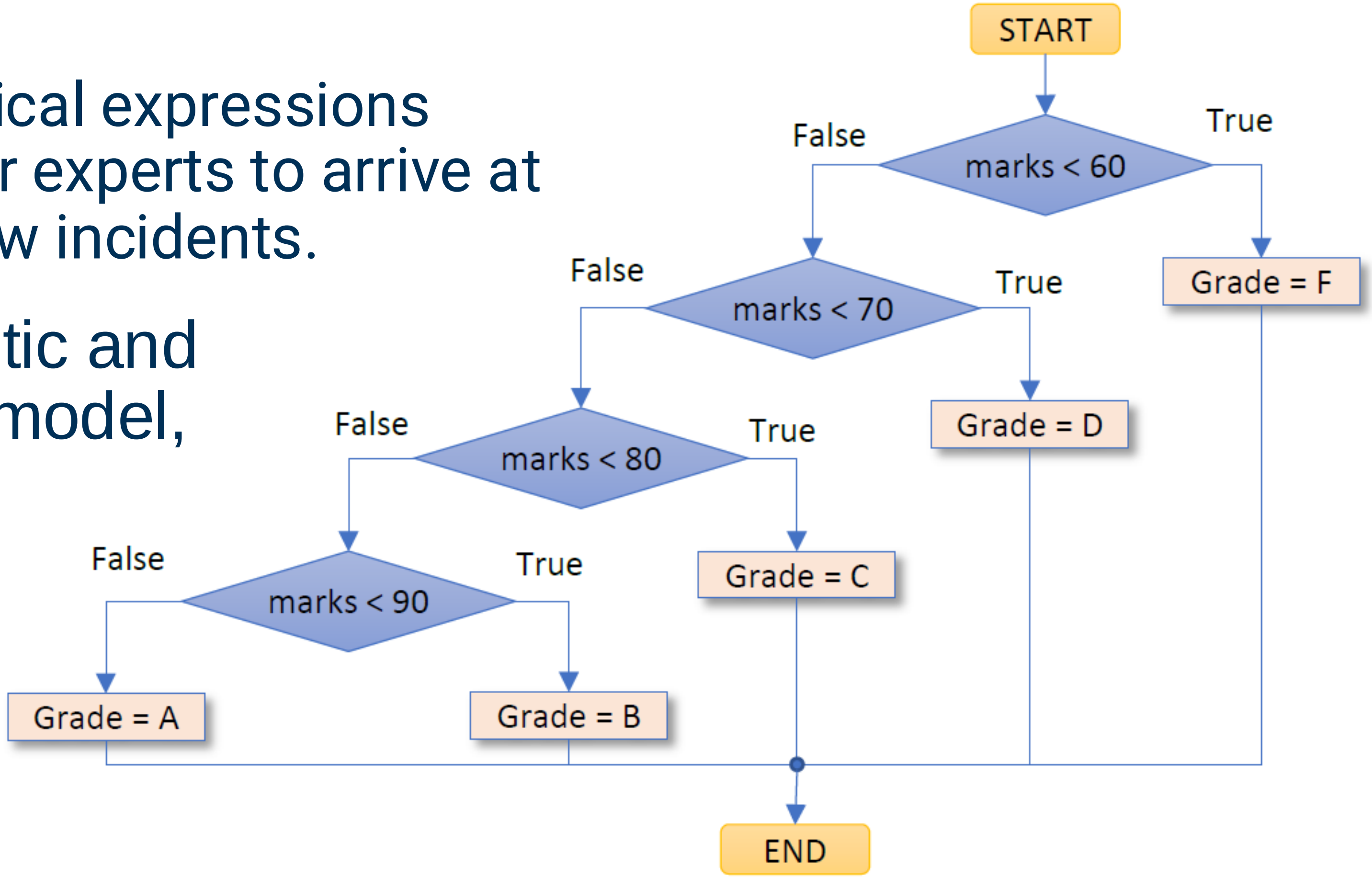
- Learning Model: A class of functions in which an algorithm searches to find one that best estimates the mapping between domain data and target output. On the implementation level, a model can have either of the following two related meanings:
  a) The structure (usually a graph) that expresses the computational process of how a prediction will be computed,
  b) The weights and biases of the computational structure which are determined by learning/training.

- Training Algorithm: The algorithm which trains the model by updating its parameters based on the training dataset to improve its mapping performance.

Rule-based Classification

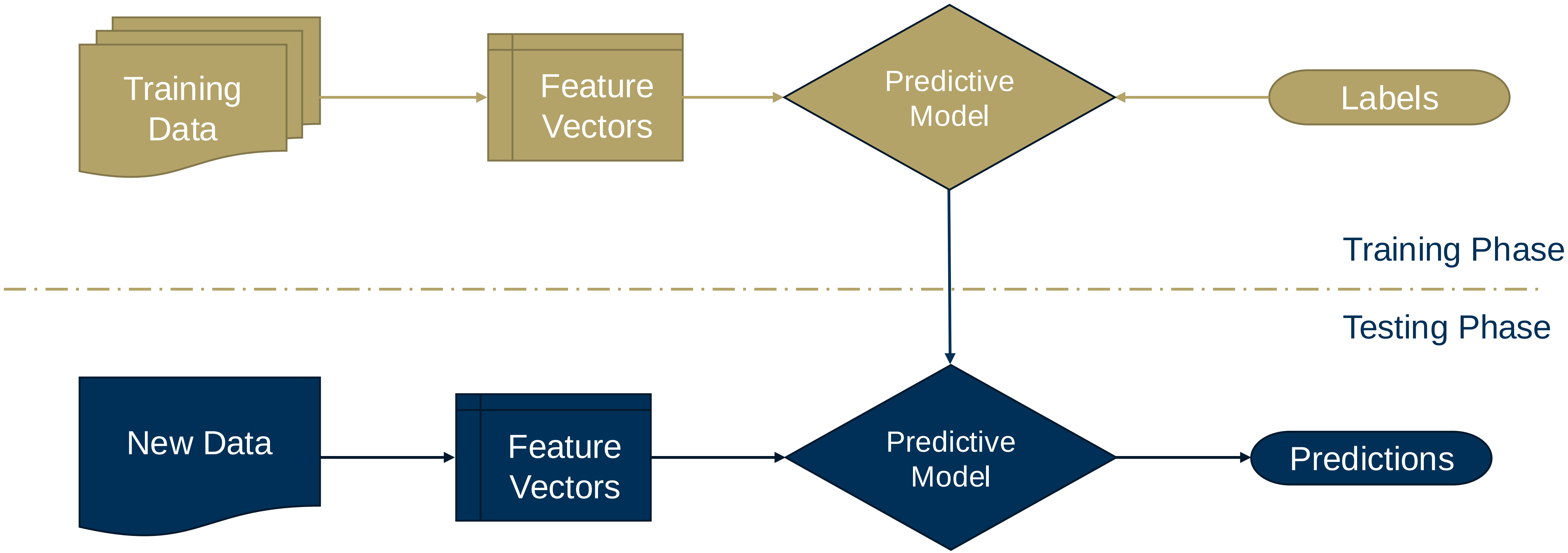
- A set of rules and mathematical expressions determined by subject matter experts to arrive at classification decision on new incidents.
- The mathematical (arithmetic and logic) expressions are the model, but it is a hardwired model!

So, your first algorithm in programming 101 was probably a classifier!



Overview

Predictor Classifier Modeling



Overview

Types of Classification

- Case-based vs. Model-based
- Feature-based vs. End-to-end
- Binary, Multi-class, Multi-label, Multi-output

Case-based vs Model-based

- Case-based:
  - Practically, lazy learners don't do any learning! Simply accumulate data and do not process it until at query time to predict class of new data.
  - New data gets label of most related data in the stored training data.
  - Example: *k*-nearest neighbor, Case-based reasoning
- Model-based:
  - Eager learners construct a classification model based on the given training data before receiving new data for classification.
  - Eager learners take a long time in training the model, and almost no time to predict.
  - Example: Decision Tree, Naive Bayes, Artificial Neural Networks

Feature-based vs End-to-End

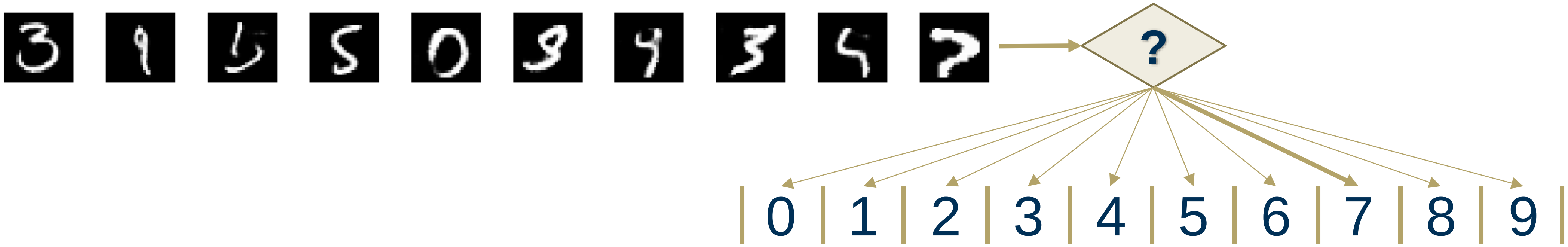
- Feature-based:
  - Must extract a set of features (feature vector) for each data item in raw data set.
  - The training set is the feature vectors, not the raw data.
  - Feature vectors are typically extracted using hand-coded pre-processing operations.
- End-to-end:
  - Training is performed on the raw data set directly.
  - Model structure learns characterizing features implicitly
  - No pre-processing is required, but learning typically takes longer than feature-based

Binary Classification

- Binary classifiers distinguish between two classes
- Also distinguish between a specific class and the rest of classes
- Can be considered a subcase of multi-class classification (number of classes = 2)
- Some classifiers are inherently binary by design
    - SVM
    - Linear classifiers

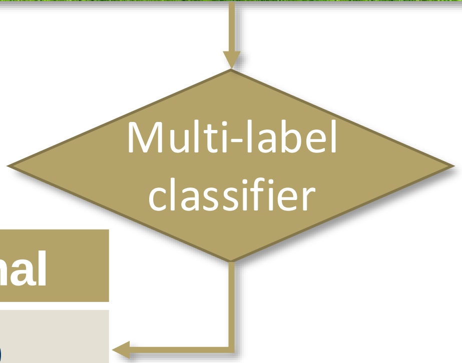
Multi-class Classification

- Also called *multinomial* classification
- Some classifiers are inherently capable of multi-class classification
  - Example: Random Forest, naïve Bayes
  - For these, binary classification is simply a sub-case
- Others are only binary by design
  - Example: SVM and Linear classifiers
  - Various strategies exist to combine multiple such classifiers to perform multi-class classification



Multi-label Classification

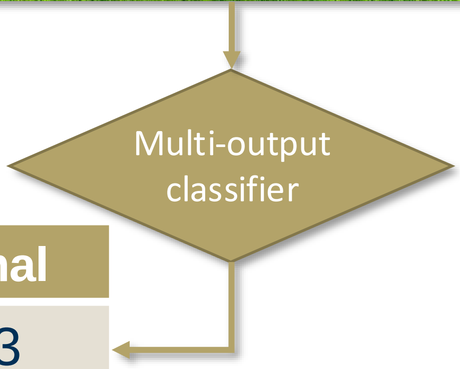
- Given a single instance (e.g., picture), classifier outputs multiple class labels (predictions)
- Output is a fixed set of labels where each label gets a binary value: Yes/No
- Suitable for detecting multiple objects of interest in the same single input incident



House	Tree	Beach	Cloud	Mountain	Animal
Yes	Yes	No	Yes	No	No

Multi-output Classification

- The general case of multi-label classification
- Fixed set of labels but a label's value is not limited to the binary set.
- For example, a multi-label classifier that gives likelihood values instead of yes/no

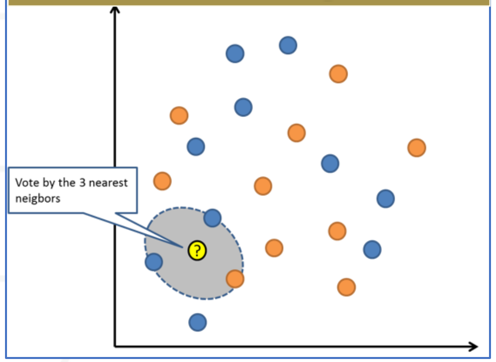


House	Tree	Beach	Cloud	Mountain	Animal
0.92	0.89	0.06	0.95	0.04	0.03

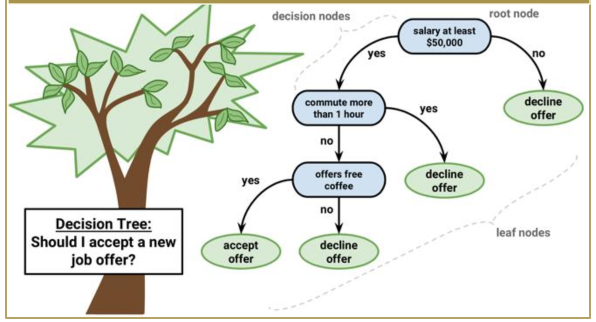
Overview

In This Course..

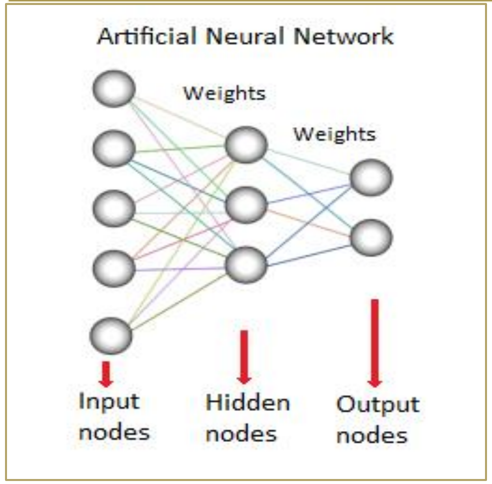
Nearest Neighbor



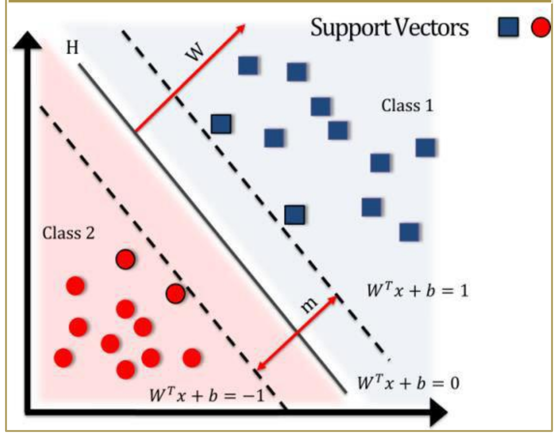
Decision Trees



Artificial Neural Networks



Support Vector Machines



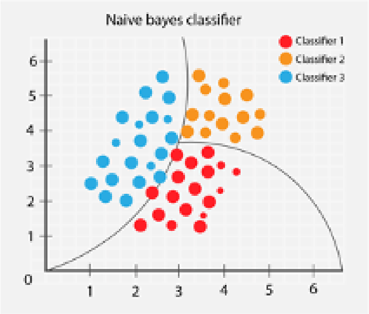
Naive Bayes Classifier

Likelihood   Class Prior Probability

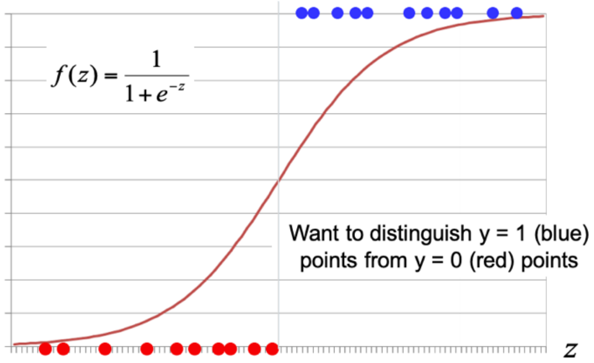
$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Posterior Probability   Predictor Prior Probability

$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$



Logistic Regression



Overview

In This Course..

Nearest Neighbor

- Overview
- Algorithm
- Distance Functions
- Choosing the value of k
- Example
- Standardized Data and Distance

Naïve Bayes

Logistic Regression

Decision Trees

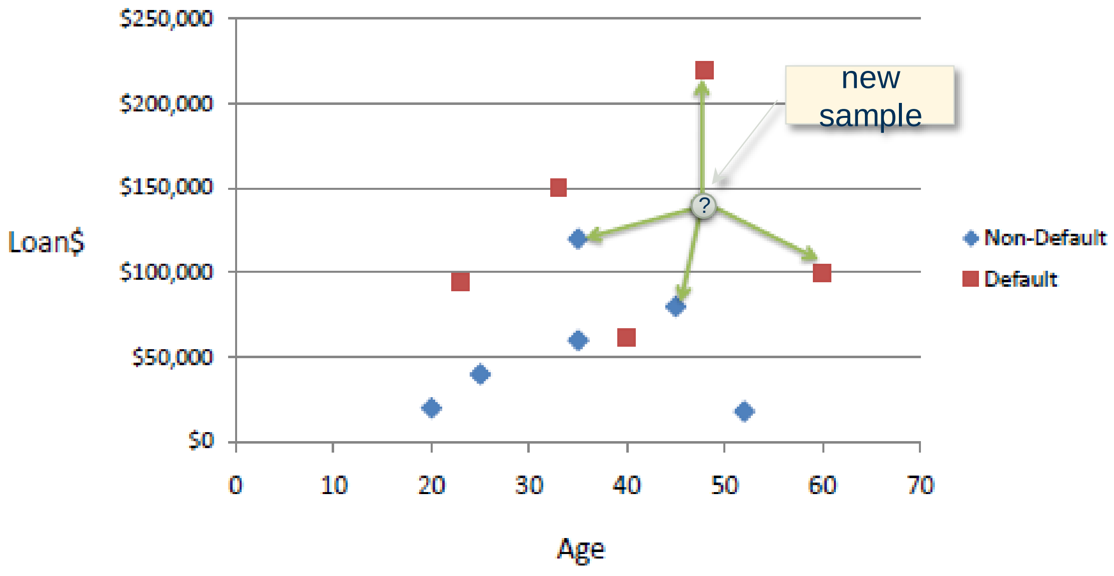
Support Vector Machines

Artificial Neural Networks

k – Nearest Neighbor (kNN) Classifier

Overview

- case-based classification
- Classifies new samples based on similarity to existing cases.
- Similarity is measured using a *distance metric*.
- k-NN has been used in statistical estimation and pattern recognition since the beginning of 1970’s
- Non-parametric technique (the only parameter is k)

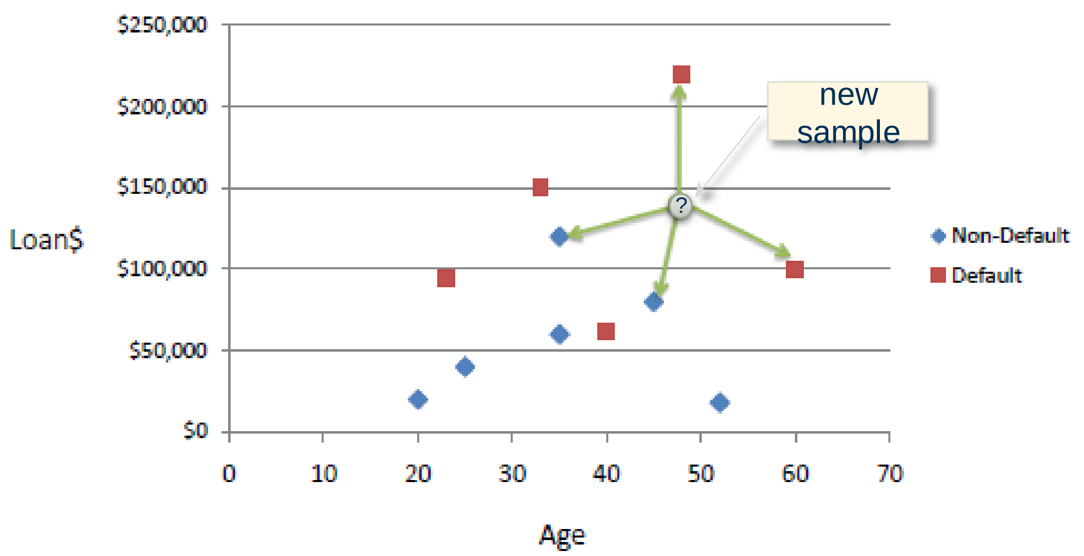


Case #	Age	Loan	Default
1	25	$40,000	N
2	35	$60,000	N
3	45	$80,000	N
4	20	$20,000	N
5	35	$120,000	N
6	52	$18,000	N
7	23	$95,000	Y
8	40	$62,000	Y
9	60	$100,000	Y
10	48	$220,000	Y
11	33	$150,000	Y

k – Nearest Neighbor (kNN) Classifier

Algorithm

- Given a dataset of samples, and a new sample to be classified,
- Distance metric is calculated over all stored samples in dataset to measure distance between new sample and each existing sample
- Nearest k samples (having smallest distances) are selected as nearest neighbors
- New sample is assigned the class label of the majority of those k nearest neighbors.
- A special case is when $k = 1$, then the case is simply assigned to the class of its nearest neighbor.
  Less computation but variance is high



How to measure distance?

Continuous variables

- *Euclidean:* $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$
- *Manhattan:* $\sum_{i=1}^{k}|x_i - y_i|$
- *Minkowski:*

$$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$$

Categorical variables

- *Hamming* Distance

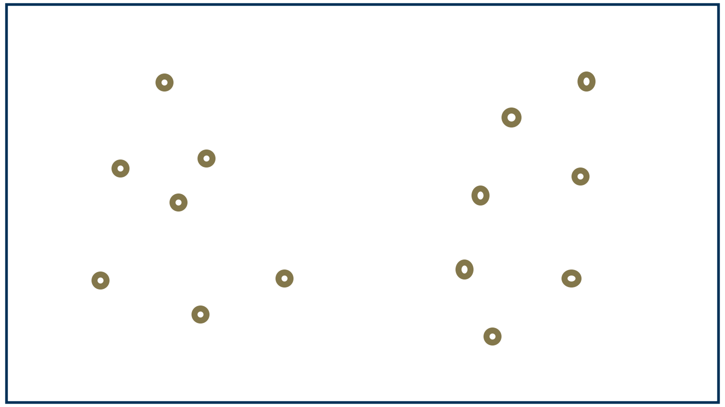
$$D_H = \sum_{i=1}^{k}|x_i - y_i|$$

$$\text{where } |x_i - y_i| = \begin{cases} 0 & if\ x_i = y_i \\ 1 & otherwise \end{cases}$$

k – Nearest Neighbor (kNN) Classifier

Choosing k

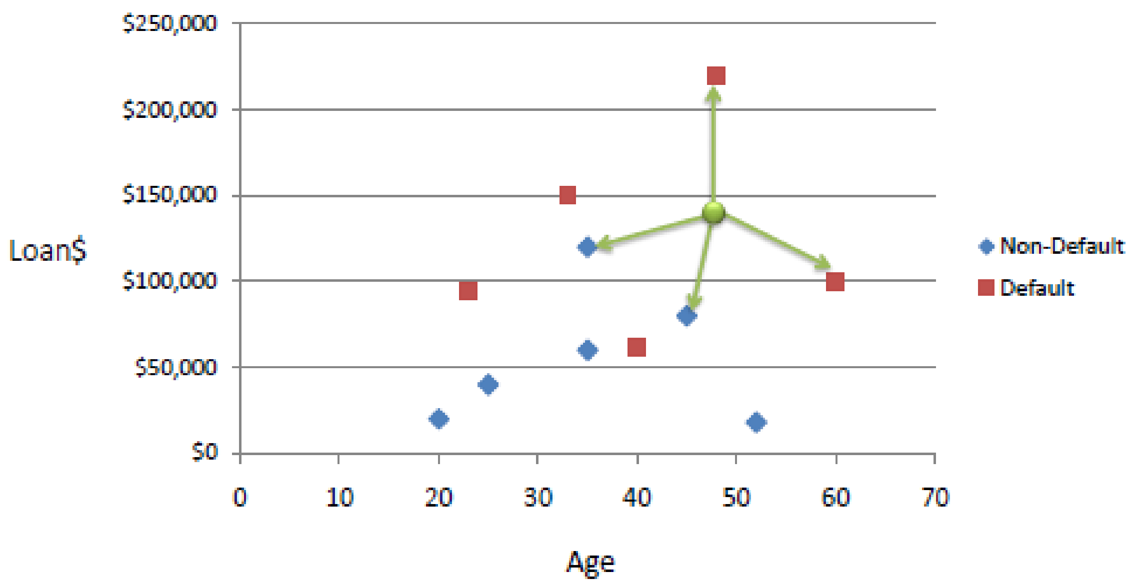
- Best done by first inspecting the data.
- In general, a large k value is more precise as it reduces the overall error but there is no guarantee.
- **Cross-validation** is another way by using an independent dataset to try multiple k values and select one with least overall error



k – Nearest Neighbor (kNN) Classifier
Example

- Classifying (predicting) whether customers may default or not.
- **Age** and **Loan** are two numerical features and **Default** is the target.

#	Age	Loan	Default
1	25	$40,000	N
2	35	$60,000	N
3	45	$80,000	N
4	20	$20,000	N
5	35	$120,000	N
6	52	$18,000	N
7	23	$95,000	Y
8	40	$62,000	Y
9	60	$100,000	Y
10	48	$220,000	Y
11	33	$150,000	Y



k – Nearest Neighbor (kNN) Classifier
Example

- Classify (predict) a new case
$$Age = 48$$
$$Loan = \$142{,}000$$
using Euclidean distance.
- The nearest instance is #11 at distance:
$$D = \sqrt{(48 - 33)^2 + (142{,}000 - 150{,}000)^2}$$
$$D = 8000.01$$
- If $k = 1$, then the classification is the label of instance #11 which is Y.
- If $k = 3$, then the nearest three neighbors are considered $\{11, 5, 9\}$, and the classification is also Y. Why?

#	Age	Loan	Default	Distance
1	25	$40,000	N	102000
2	35	$60,000	N	82000
3	45	$80,000	N	62000
4	20	$20,000	N	122000
5	35	$120,000	N	22000
6	52	$18,000	N	124000
7	23	$95,000	Y	47000
8	40	$62,000	Y	80000
9	60	$100,000	Y	42000
10	48	$220,000	Y	78000
11	33	$150,000	Y	8000

$k-$ Nearest Neighbor (kNN) Classifier

Data Normalization

- Distance measure can be biased when:
  - variables have different measurement scales, or
  - there is a mixture of numerical and categorical variables.
- For example:
  - Loan is in dollars, while age is in years,
  - Loan has a higher influence on the distance since it has a much wider range in values.
- One solution: standardize the data by normalizing each column using the formula:

$$x_{i,j} = \frac{x_{i,j} - \min(\boldsymbol{x}_{:,j})}{\max(\boldsymbol{x}_{:,j}) - \min(\boldsymbol{x}_{:,j})}$$

#	Age	Loan	Default	Distance
1	25	$40,000	N	102000
2	35	$60,000	N	82000
3	45	$80,000	N	62000
4	20	$20,000	N	122000
5	35	$120,000	N	22000
6	52	$18,000	N	124000
7	23	$95,000	Y	47000
8	40	$62,000	Y	80000
9	60	$100,000	Y	42000
10	48	$220,000	Y	78000
11	33	$150,000	Y	8000

#	Age	Loan	Default	Distance
1	0.125	0.11	N	0.7652
2	0.375	0.21	N	0.5200
3	0.625	0.31	N	**0.3160**
4	0.000	0.01	N	0.9245
5	0.375	0.50	N	**0.3428**
6	0.800	0.00	N	0.6220
7	0.075	0.38	Y	0.6669
8	0.500	0.22	Y	0.4437
9	1.000	0.41	Y	**0.3650**
10	0.700	1.00	Y	0.3861
11	0.325	0.65	Y	0.3771

Overview

In This Course..

- Nearest Neighbor
- Naïve Bayes
  - Overview
  - Example: a) single feature
  - Example: b) two or more features
- Logistic Regression
- Decision Trees
- Support Vector Machines
- Artificial Neural Networks

Naïve Bayes
Overview

- Used for both classification and regression
- Assumes that all features in a dataset are:
  - Equally important (no weights), and
  - Independent
- Requires very small computational power, thus works fast even with large data
- Key terms:
  - (Class) Prior Probability: Proportion of dependent variable (target) in the data set.
  - Likelihood: Probability of classification class of a given observation in presence of some other variable
  - Marginal likelihood: Proportion of independent variable (predictor) in the data set

Naïve Bayes

Case Study: Single Feature

- Weather condition and corresponding target variable "Play"
- Objective: classify (predict) whether to play or not based on weather condition
- Steps:
  - Step 1: Convert the data set into a frequency table (also called contingency table)
  - Step 2: Create Likelihood table.
  - Step 3: Use Naive Bayesian equation to calculate the posterior probability for each class.

The class with the highest posterior probability is the outcome of prediction.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Naïve Bayes

Case Study: Single Feature

- Step 1 and Step 2

Data set

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency (contingency) Table

	No	Yes	Total
Overcast	0	4	4
Rainy	3	2	5
Sunny	2	3	5
Total	5	9	**14**

$$P(rainy) = \frac{5}{14}$$

$$P(Yes) = \frac{9}{14} \qquad P(rainy \mid Yes) = \frac{2}{9}$$

$$P(No) = \frac{5}{14} \qquad P(rainy \mid No) = \frac{3}{5}$$

$$P(Yes \mid rainy) = ? \qquad P(No \mid rainy) = ?$$

Naïve Bayes

Case Study: Single Feature

	No	Yes	Total
Overcast	0	4	4
Rainy	3	2	5
Sunny	2	3	5
Total	5	9	14

The terms ***Likelihood***, ***Marginal Likelihood***, and ***Prior Probability*** (or ***Class Prior Probability***, as it is related to classes "Yes" or "No") that were mentioned above are shown below:

Likelihood (lime color divided by olive drab color)
Example: P (*Overcast* | "*Yes*") = 4/9

Marginal Likelihood or **Predictor Prior Probabilities** (red color)
Example: P (*Sunny*) = 5/14

Likelihood = *P* (*Feature* | Class)

Weather value	No	Yes	Total
Overcast	0	4/14	4/14
Rainy	3/14	2/14	5/14
Sunny	2/14	3/14	5/14
Total	5/14	9/14	N

Feature value	Class A	Class B	Total
Value A	a/N	d/N	(a+d)/N
Value B	b/N	e/N	(b+e)/N
Value C	c/N	f/N	(c+f)/N
Total	(a+b+c)/N	(d+e+f)/N	N

Prior Probabilities or **Class Prior Probabilities** (olive drab color)
Example: P (*No*) = 5/14

Prior Probabilities = *P (Class)*

Marginal Likelihood = *P (Feature)*

Naïve Bayes

Case Study: Single Feature

- Step 3:
  - Use Bayes' Formula to calculate the *posterior probability* for each class.
  - The class with the highest posterior probability is the outcome of prediction.

Likelihood
P(feature|class)

Class Prior Probability
P(class)

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})}$$

Posterior Probability
P(class|feature)

Predictor Prior Probability
P(feature)

Naïve Bayes

Case Study: Single Feature

- Step 3:
  - Use Bayes' Formula to calculate the *posterior probability* for each class.
  - The class with the highest posterior probability is the outcome of prediction.

Likelihood
P(feature|class)

Class Prior Probability
P(class)

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})}$$

Posterior Probability
P(class|feature)

Predictor Prior Probability
P(feature)

$$P(Yes \mid rainy) = \frac{P(rainy \mid Yes)\, P(Yes)}{P(rainy)} = \frac{\frac{2}{9} \times \frac{9}{14}}{\frac{5}{14}} = \frac{2}{5} = 0.4$$

OLIVES @GeorgiaTech

Georgia Tech

Naïve Bayes

Case Study: Single Feature

- Step 3:
  - Use Bayes' Formula to calculate the *posterior probability* for each class.
  - The class with the highest posterior probability is the outcome of prediction.

Likelihood
P(feature|class)

Class Prior Probability
P(class)

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})}$$

Posterior Probability
P(class|feature)

Predictor Prior Probability
P(feature)

$$P(No\,|\,rainy) = \frac{P(rainy\,|\,No)\,P(No)}{P(rainy)} = \frac{\frac{3}{5} \times \frac{5}{14}}{\frac{5}{14}} = \frac{3}{5} = 0.6 = 1 - 0.4$$

Naïve Bayes

Case Study: Two or More Feature

- Let’s assume we have the additional feature “**Wind**”
- Let’s assume we want to predict the class for the following case:

  Wind = Moderate

  Weather = Sunny
- Thus, we want to estimate

$$P(play = Yes|x_1 = sunny, x_2 = moderate)$$

$$P(play = No|x_1 = sunny, x_2 = moderate)$$

And choose the large value

Weather	Wind	Play
Sunny	Strong	No
Overcast	Weak	Yes
Rainy	Moderate	Yes
Sunny	Weak	Yes
Sunny	Weak	Yes
Overcast	Weak	Yes
Rainy	Strong	No
Rainy	Strong	No
Sunny	Weak	Yes
Rainy	Weak	Yes
Sunny	Strong	No
Overcast	Weak	Yes
Overcast	Weak	Yes
Rainy	Moderate	No

	No	Yes	Total
Overcast	0	4	4
Rainy	3	2	5
Sunny	2	3	5
Total	5	9	**14**

	No	Yes	Total
Strong	4	0	4
Weak	0	8	8
Moderate	1	1	2
Total	5	9	14

Naïve Bayes

Case Study: Two or More Feature

$$P(yes|sunny, moderate) = \frac{P(sunny, moderate|Yes)P(Yes)}{P(sunny, moderate)}$$

Here is the "naive" in Naïve Bayes:

- It assumes that the features are conditionally independent
- If x_1, x_2 are conditionally independent, then:

$$P(x_1, x_2|y) = P(x_1|y)P(x_2|y)$$

Hence,

$$P(yes|sunny, moderate) = \frac{P(sunny|Yes)P(moderate|Yes)P(Yes)}{P(sunny, moderate)}$$

And

$$P(No|sunny, moderate) = \frac{P(sunny|Np)P(moderate|No)P(No)}{P(sunny, moderate)}$$

Weather	Wind	Play
Sunny	Strong	No
Overcast	Weak	Yes
Rainy	Moderate	Yes
Sunny	Weak	Yes
Sunny	Weak	Yes
Overcast	Weak	Yes
Rainy	Strong	No
Rainy	Strong	No
Sunny	Weak	Yes
Rainy	Weak	Yes
Sunny	Strong	No
Overcast	Weak	Yes
Overcast	Weak	Yes
Rainy	Moderate	No

Naïve Bayes

Case Study: Two or More Feature

$$P(yes|sunny, moderate) \propto P(sunny|yes)P(moderate|yes)P(yes)$$

And

$$P(No|sunny, moderate) \propto P(sunny|No)P(moderate|No)P(No)$$

$$P(Yes|sunny, moderate) = \frac{\frac{3}{9} \times \frac{1}{9} \times \frac{9}{14}}{constant} = \frac{\frac{1}{42}}{constant}$$

$$P(No|sunny, moderate) = \frac{\frac{2}{5} \times \frac{1}{5} \times \frac{5}{14}}{constant} = \frac{\frac{1}{35}}{constant}$$

Therefore, the Naïve Bayes classifier will result in **No**

Weather	Wind	Play
Sunny	Strong	No
Overcast	Weak	Yes
Rainy	Moderate	Yes
Sunny	Weak	Yes
Sunny	Weak	Yes
Overcast	Weak	Yes
Rainy	Strong	No
Rainy	Strong	No
Sunny	Weak	Yes
Rainy	Weak	Yes
Sunny	Strong	No
Overcast	Weak	Yes
Overcast	Weak	Yes
Rainy	Moderate	No

	No	Yes	Total
Overcast	0	4	4
Rainy	3	2	5
Sunny	2	3	5
Total	5	9	**14**

	No	Yes	Tota l
Strong	4	0	4
Weak	0	8	8
Moderate	1	1	2
Total	5	9	14

Naïve Bayes

Case Study: Two or More Feature

- Using conditional independence of feature vector $\boldsymbol{x} = \left[x_1, x_2, \dots, x_p\right]^T$ given a class labels vector $y = [y_1, y_1, \dots, y_k]^T$, the posterior probability of a class y_k given the feature vector $\boldsymbol{x}$ is:

$$P(y_k|\boldsymbol{x}) = \frac{P(x_1|y_k)P(x_2|y_k)\dots P(x_p|y_k)P(y_k)}{P(\boldsymbol{x})}$$

$$P(y_k|\boldsymbol{x}) \propto P(x_1|y_k)P(x_2|y_k)\dots P(x_p|y_k)P(y_k)$$

Then, the class can be found as:

$$class = \arg\max_i P(y_i|\boldsymbol{x})$$

Naïve Bayes

Types of Classifiers

Different assumptions about $P(\boldsymbol{x}_{:,i}|y_k)$

- Categorical
  - $P(\boldsymbol{x}_{:,i} = a | y_k = b) = \frac{N_{a,b}+\alpha}{N_b+\alpha n_i}$
  - where
    - $N_{a,b}$: Number of samples with $\boldsymbol{x}_{:,i} = a$ and $y_k = b$
    - N_b: Number of samples with $y_k = b$
    - n_i: The number of categories in $\boldsymbol{x}_{:,i}$
- Gaussian:
  - $P(\boldsymbol{x}_{:,i}|y_k) = \frac{1}{\sqrt{2\pi\sigma_{y_k}^2}} \exp\left(-\frac{\left(\boldsymbol{x}_{:,i}-\mu_{y_k}\right)^2}{2\sigma_{y_k}^2}\right)$
- Bernoulli
- Multinomial
- Complement
- More details can be found here: https://scikit-learn.org/stable/modules/naive_bayes.html#naive-bayes

Naïve Bayes

Pros and Cons

- **Pros:**
  - Fast
  - Performance (assuming independence)
  - Better at categorical input
- **Cons:**
  - Zero-frequency
  - Assumption of independence

Naïve Bayes

Common Applications

- **Real time Prediction:** Naive Bayes is an eager learning classifier and it is fast. Thus, it could be used for making predictions in real time.
- **Multi-class Prediction:** This algorithm is also well known for multi-class prediction feature. Here we can predict the probability of multiple classes of target variable.
- **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
- **Recommendation System:** Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

Overview

In This Course..

- Nearest Neighbor
- Naïve Bayes
- Logistic Regression
  - Overview
  - Example
- Decision Trees
- Support Vector Machines
- Artificial Neural Networks

Appendix A: Notations

- x_i: a single feature
- $\boldsymbol{x}_i$: feature vector (data sample)
- $\boldsymbol{X}$: matrix of feature vectors (dataset)
- N: number of data samples
- m: degree of polynomial
- P: number of features in a feature vector
- θ_i: a single model coefficient (parameter)
- $\boldsymbol{\theta}$: coefficient vector
- ε: error margin
- α: learning rate
- γ: bias factor
- Bold letter/symbol: vector
- Bold capital letters/symbol: matrix