Data Mining & Machine Learning

Yong Zheng

Illinois Institute of Technology
Chicago, IL, 60616, USA

ILLINOIS TECH
College of Computing

Schedule

- Supervised & Unsupervised Learning
- Supervised Learning: Classification
- Classification Algorithms
  - KNN Classifier
  - Naïve Bayes Classifier

Important Notes

- Emphasis: understanding!!!
- You must understand the techniques
  - What it is
  - What problems it can solve
  - In which situations we should use them
  - Any limitations or requirements to use them
  - How to evaluate them

Schedule

- Supervised & Unsupervised Learning
- Supervised Learning: Classification
- Classification Algorithms
  - KNN Classifier
  - Naïve Bayes Classifier

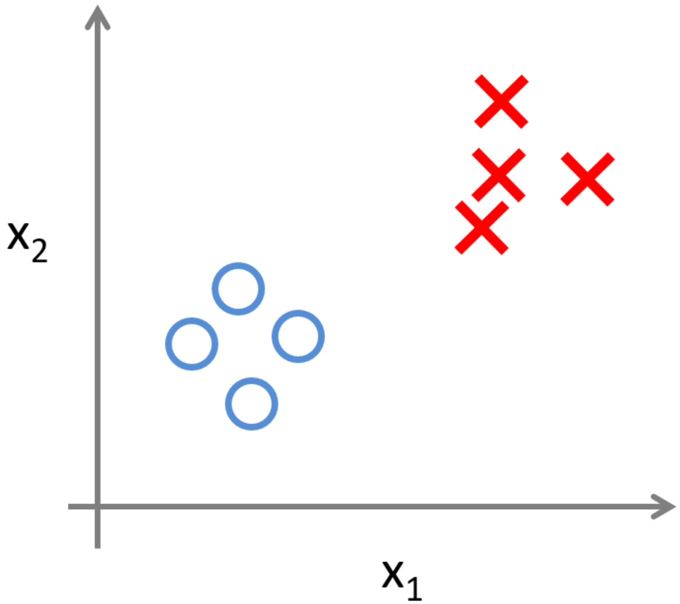
Supervised v.s. Unsupervised Learning

- Supervised Learning: infer a (predictive) function from data associated with pre-defined targets/classes/labels
  Example: group objects by predefined labels
  Goal: Learn a model from labelled data (with multiple features) for future predictions
  Outcomes: We know outcomes: the predefined labels
  Evaluation: error/accuracy, and other more metrics
  Data Mining Task: Classification

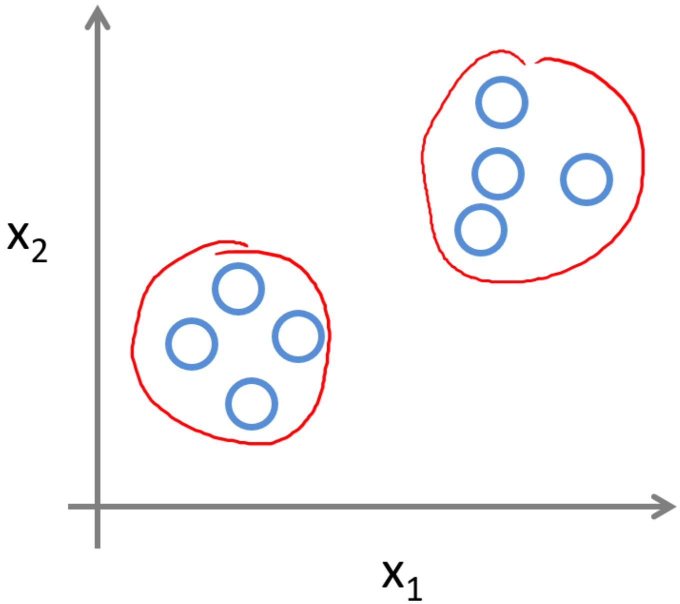
- Unsupervised Learning: discover or describe underlying structure from unlabelled data
  Example: group objects by multiple features
  Goal: Learn the structure from unlabelled data (with multiple features)
  Outcomes: We do not know the outcomes
  Evaluation: No clear performance or evaluation methods
  Data Mining Task: Clustering

Supervised v.s. Unsupervised Learning

Supervised Learning	Unsupervised Learning



Example: Classification



Example: Clustering

Supervised v.s. Unsupervised Learning

Machine Learning Algorithms *(sample)*

	Unsupervised	Supervised
Continuous	• Clustering & Dimensionality Reduction ◦ SVD ◦ PCA ◦ K-means	• Regression ◦ Linear ◦ Polynomial • Decision Trees • Random Forests
Categorical	• Association Analysis ◦ Apriori ◦ FP-Growth • Hidden Markov Model	• Classification ◦ KNN ◦ Trees ◦ Logistic Regression ◦ Naive-Bayes ◦ SVM

Supervised Learning: Linear Regression

- We have knowledge: values in y
- We have factors or features: x variables
- We need to split data into training and testing
- We learned the model from training, and evaluate it on the testing set
- We do have truth in testing test and predictions for test set, as well as evaluation metrics: RMSE, MAE
- Have a general problem in supervised learning: overfitting

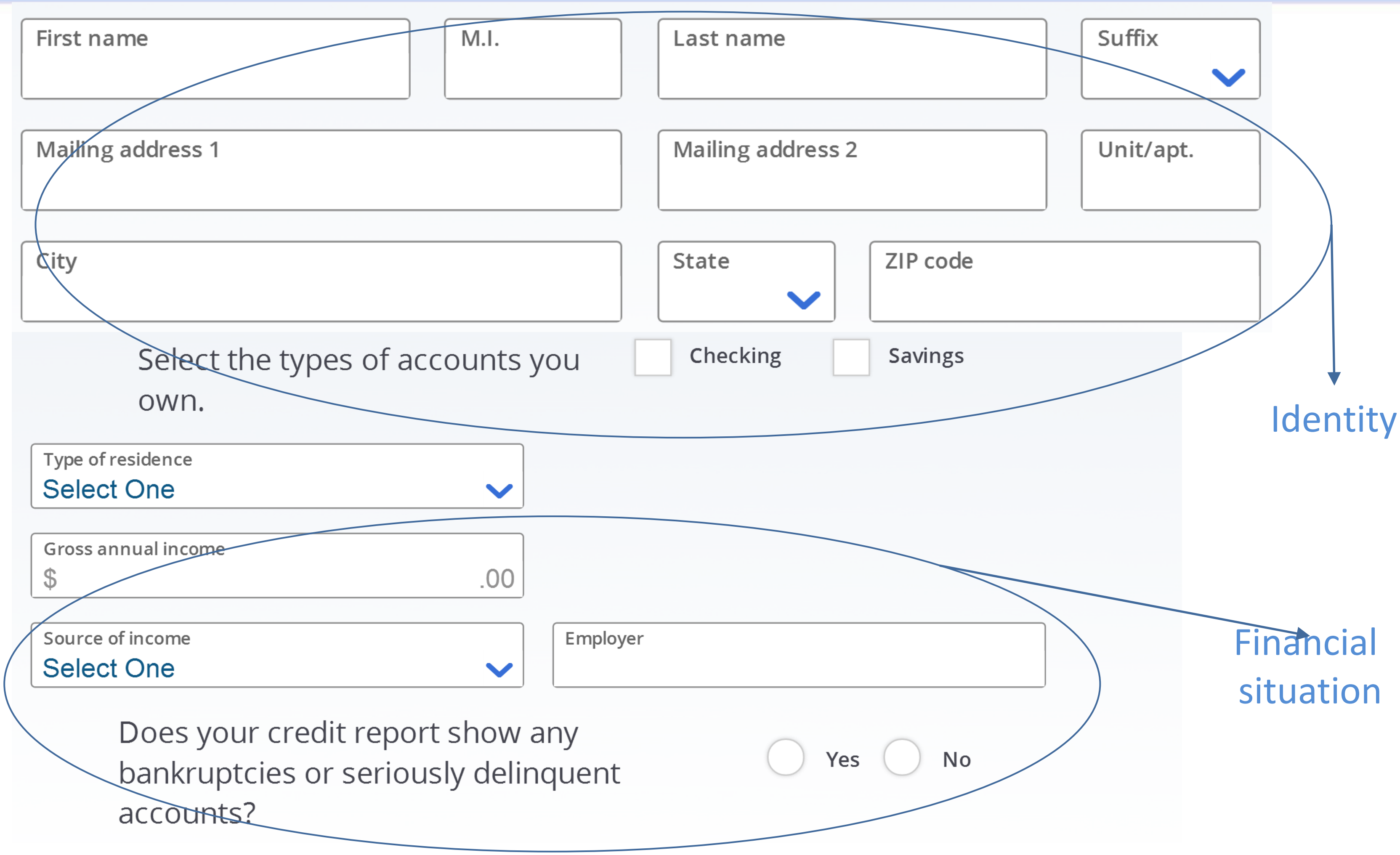
Schedule

- Supervised & Unsupervised Learning
- Supervised Learning: Classification
- Classification Algorithms
  - KNN Classifier
  - Naïve Bayes Classifier

Supervised Learning: Classification

- Classification: a supervised way to group objects
  - We must have predefined labels
  - We must have knowledge: we know some instances are labeled by predefined classes/labels/categories
- For a Purpose of Prediction
  - To forecast or deduce the label/class based on values of features
  - Let the machines/computers think as humans
- There are many real-world applications
  - Financial Decision Making, e.g., credit card application
  - Image Processing, e.g., face recognition in cameras
  - Computer/Network Security, e.g., virus or attack detection
  - Information Retrieval, e.g., relevance of a document to a query
  - Recommender Systems, e.g., rating prediction for Amazon

Classification App: Credit Card Application



Classification App: Credit Card Application

Date Received	Card	Status of Application
05/21/15	THE AMERICAN EXPRESS BUSINESS PLATINUM CARD	Approved
07/22/15	THE GOLD DELTA SKYMILES BUSINESS CREDIT CARD	Rejected
08/19/15	PREMIER REWARDS GOLD CARD FROM AMERICAN EXPRESS	Under Review

Classification App: Credit Card Application

Terminologies in Classification

Features

classes

Age	Gender	Status	Income	Rent	Classes
27	Female	Student	$15,000	$800	Approved
32	Male	Part-time	$8,000	$400	Rejected
29	Male	Full-time	$50,000	$1200	?

Knowledge

Unseen data

Each row with features values is named as example or instance

Classification → Learn from the knowledge (examples with unknown labels)
build predictive models to predict the unknown examples

Classification

- Classification Tasks
- Standard Classification Process
- Evaluation: How could we know it is good or bad
- General Problem: overfitting
- Algorithms: How to perform classification tasks

Classification

- Classification Tasks
- Standard Classification Process
- Evaluation: How could we know it is good or bad
- General Problem: overfitting
- Algorithms: How to perform classification tasks

Classification Task

There are usually three types of classification:

1). Binary Classification
Question: Is this an apple? Yes or No.

2). Multi-class Classification
Question: Is this an apple, banana or orange?

3). Multi-label Classification
Use appropriate words to describe it:
Red, Apple, Fruit, Tech, Mac, iPhone

Color, Shape, Weight, Origin, Taste, Price, Vitamin c

Classification Task

There are usually three types of classification:

1). Binary Classification
Question: Is this an apple? Yes or No.

2). Multi-class Classification
Question: Is this an apple, banana or orange?

3). Multi-label Classification
Use appropriate words to describe it:
Red, Apple, Fruit, Tech, Mac, iPhone

We use binary classification as examples to introduce classification techniques. But most of these classification methods can handle multi-class classifications too. There are different strategies to handle multi-class classifications.

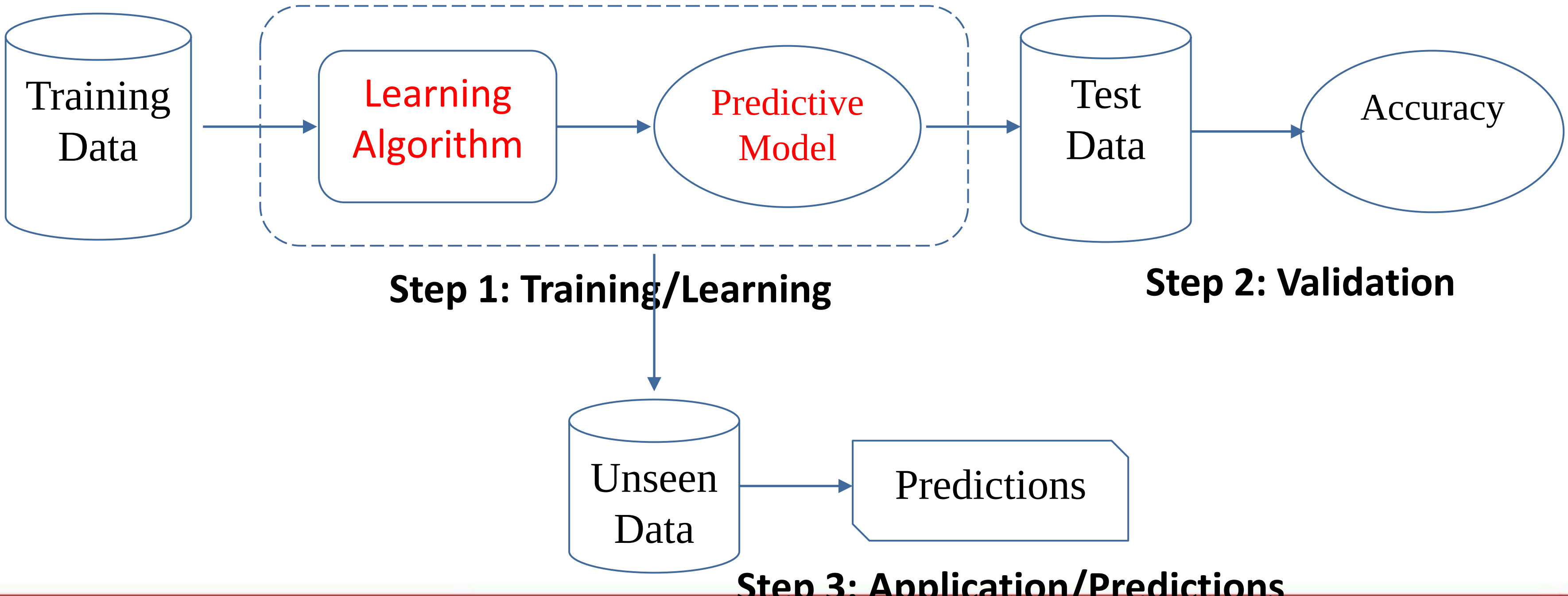
Classification

- Classification Tasks
- Standard Classification Process
- Evaluation: How could we know it is good or bad
- General Problem: overfitting
- Algorithms: How to perform classification tasks

Standard Classification Process

- Train: Learn a model using the training data
- Validation/Test: Test using test data to assess accuracy
- Application: Apply the selected model to unseen data

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}}$$



Classification

- Classification Tasks
- Standard Classification Process
- Evaluation: How could we know it is good or bad
- General Problem: overfitting
- Algorithms: How to perform classification tasks

Data Splits for Evaluations

- There are several ways to split your data for evaluations
  - Hold-out evaluation
  - N-fold cross validation
  - Leave-one-out evaluation
  - Stratified N-fold cross validation
  - …..

Data Splits for Evaluations

1). Hold-out Evaluation → If your data is large enough

Color	Weight (lbs)	Stripes	Tiger?
Orange	300	no	no
White	50	yes	no
Orange	490	yes	yes
White	510	yes	yes
Orange	490	no	no
White	450	no	no
Orange	40	no	no
Orange	200	yes	no
White	500	yes	yes
Green	560	yes	no
Orange	500	yes	?
White	50	yes	?

Training Data Set

Test Data Set

Unseen data set

Data Splits for Evaluations

1). Hold-out Evaluation → If your data is large enough

Color	Weight (lbs)	Stripes	Tiger?
Orange	300	no	no
White	50	yes	no
Orange	490	yes	yes
White	510	yes	yes
Orange	490	no	no
White	450	no	no
Orange	40	no	no
Orange	200	yes	no
White	500	yes	yes
Green	560	yes	no
Orange	500	yes	?
White	50	yes	?

Training Data Set

Validation Data Set

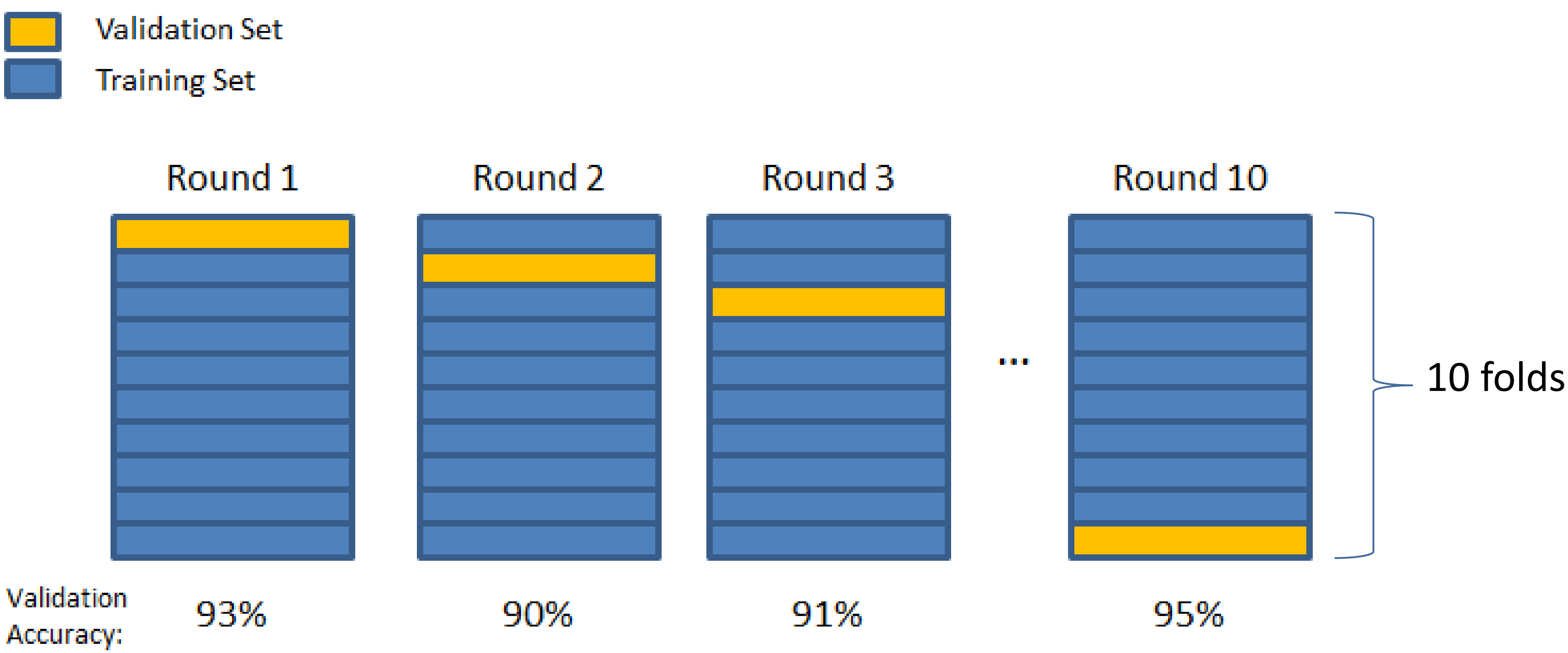
→ Training, evaluating and tunning model parameters

Test Data Set → Report results on testing set only

Unseen data set

Data Splits for Evaluations

2). N-folds Cross Evaluation → If your data is relatively small



Final Accuracy = Average(Round 1, Round 2, ...)

Data Splits for Evaluations

- Summary
  - We always suggest you to use N-fold cross validation, as long as you have enough computational power – it doesn't matter your data is large or small
  - If your computer is not powerful
    - Data is large => you can use hold-out
    - Data is small => you must use N-fold cross validation
    - No fixed rule to say data is large or small. Usually, a data set with less than 500K rows can be considered as small data
  - Common mistakes: some students run both hold-out and N-fold cross validation, and report best results

How it works: Build a Model

Instances

Features + classes

Training Data

Classification Algorithms

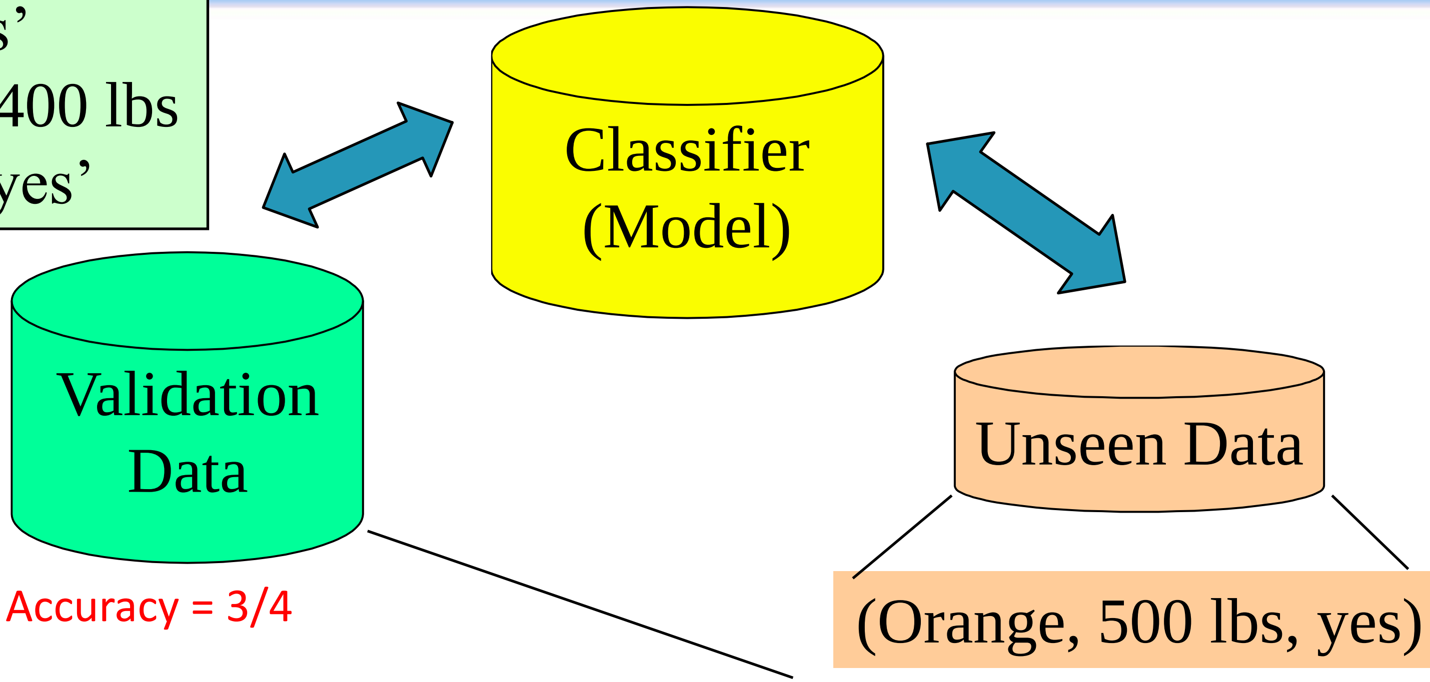
Classifier (Model)

Color	Weight	Stripes	Tiger?
Orange	300 lbs	no	no
White	50 lbs	yes	no
Orange	490 lbs	yes	yes
White	510 lbs	yes	yes
Orange	490 lbs	no	no
White	450 lbs	no	no

IF stripes = 'yes'
AND weight > 400 lbs
THEN tiger = 'yes'

How it works: Predictions

IF stripes = ‘yes’
AND weight > 400 lbs
THEN tiger = ‘yes’



Color	Weight	Stripes	Pred	Truth
Orange	40 lbs	no	no	no
Orange	200 lbs	yes	no	no
White	500 lbs	yes	yes	yes
Green	560 lbs	yes	yes	no

Tiger?

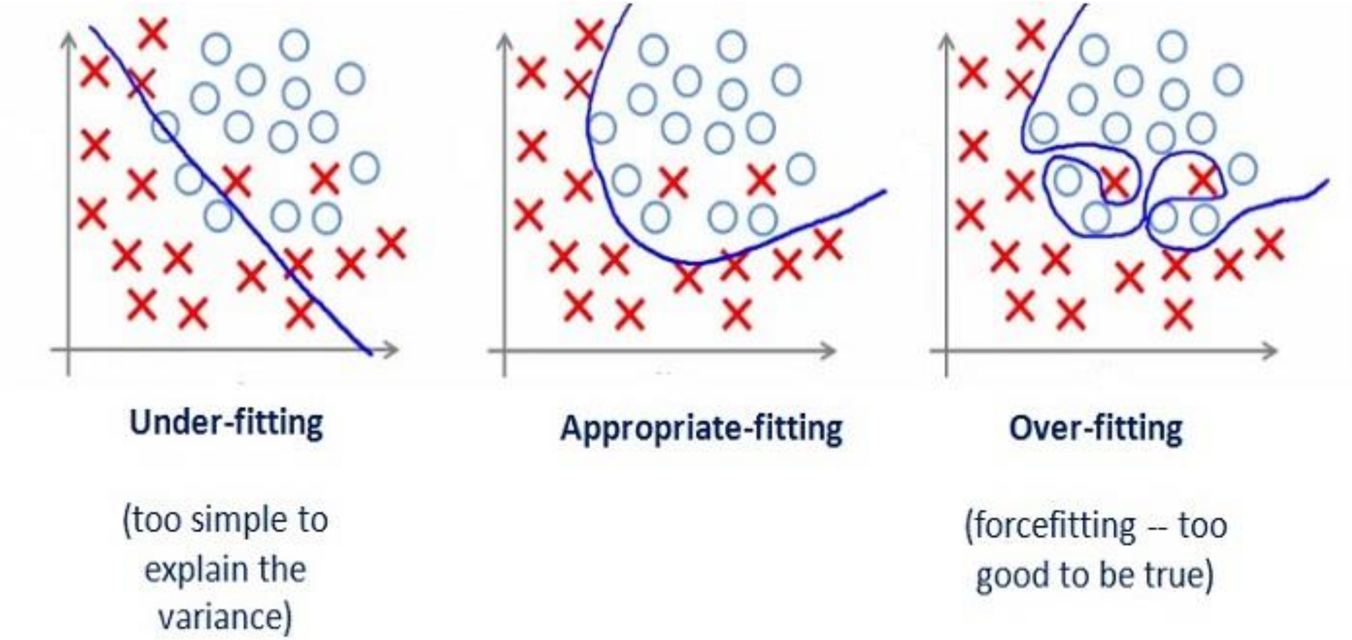
Yes

Classification

- Classification Tasks
- Standard Classification Process
- Evaluation: How could we know it is good or bad
- General Problem: overfitting
- Algorithms: How to perform classification tasks

Overfitting Problem

Problem: The model is over-trained by the training set; the performance on the testing set (such as accuracy) is significantly worse than the performance on training set



Example of over-trained: students can work on questions on the assignment well, but they may not work well on the questions in the exams.

Example: Overfitting

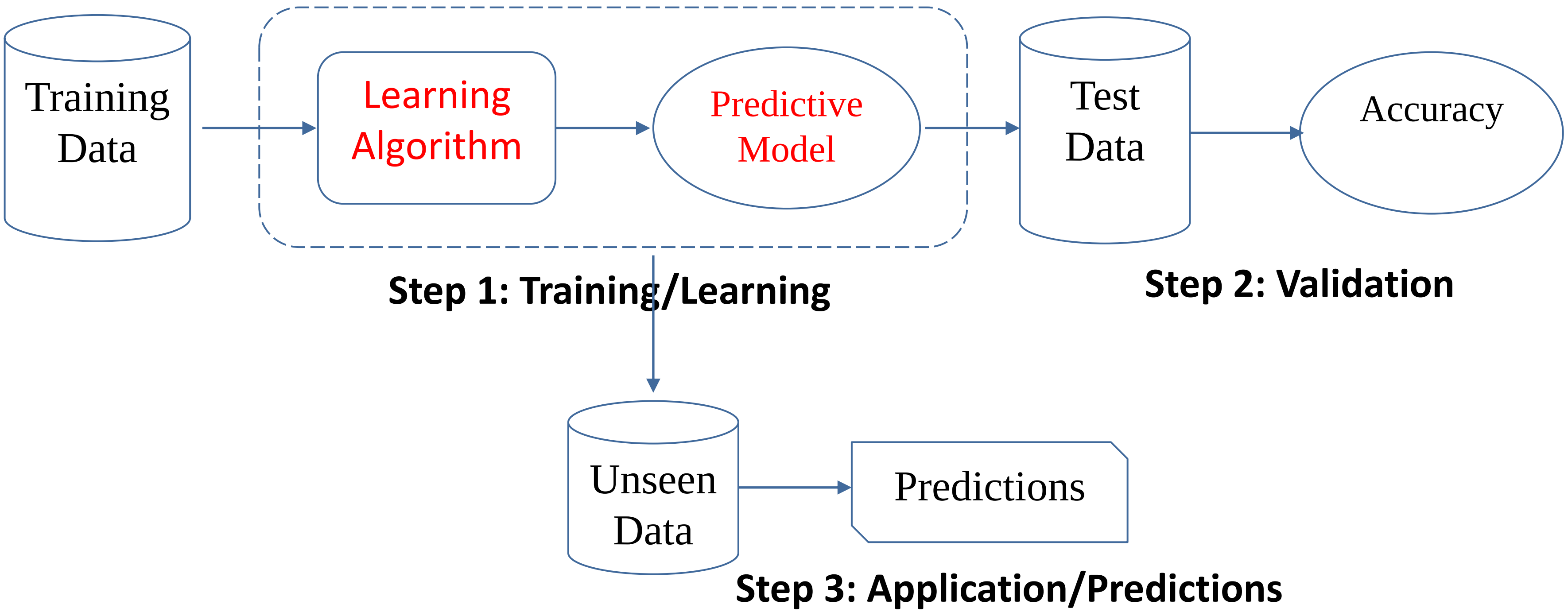
- Is there an overfitting problem?
- Linear Regression Models
  - M1: Adj-R2 = 96%, MAE = 0.36
  - M2: Adj-R2 = 98%, MAE = 0.6
- Classification Models
  - M1: Accuracy on training = 90%, testing = 85%
  - M2: Accuracy on training = 80%, testing = 85%
  - M3: Accuracy on training = 85%, testing = 60%

Classification

- Classification Tasks
- Standard Classification Process
- Evaluation: How could we know it is good or bad
- General Problem: overfitting
- Algorithms: How to perform classification tasks

Classification

- **Classification algorithm is the key component in the process**
- **They are able to learn from training and build models…**



Schedule

- Supervised & Unsupervised Learning
- Supervised Learning: Classification
- Classification Algorithms

Classification Algorithms

- Classification algorithm is the key component in the process
- They are able to learn from training and build models

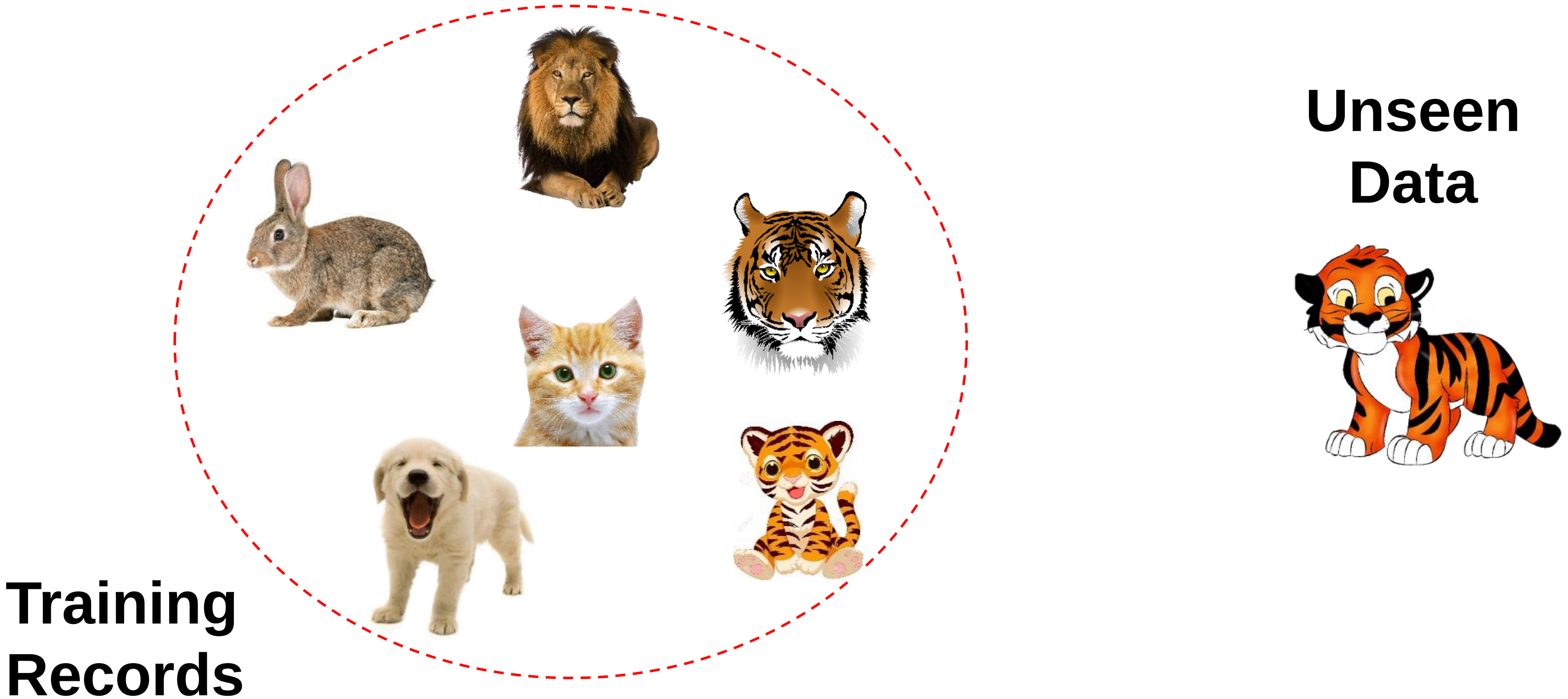
There are many (supervised) classification algorithms:

- K-nearest neighbor classifier
- Naïve Bayes classifier
- Decision tress
- Linear/Logistic regression
- Support Vector Machines
- Ensemble classifiers (e.g., random forest)
- Neural Networks
- …

Classification Algorithms: KNN Classifier

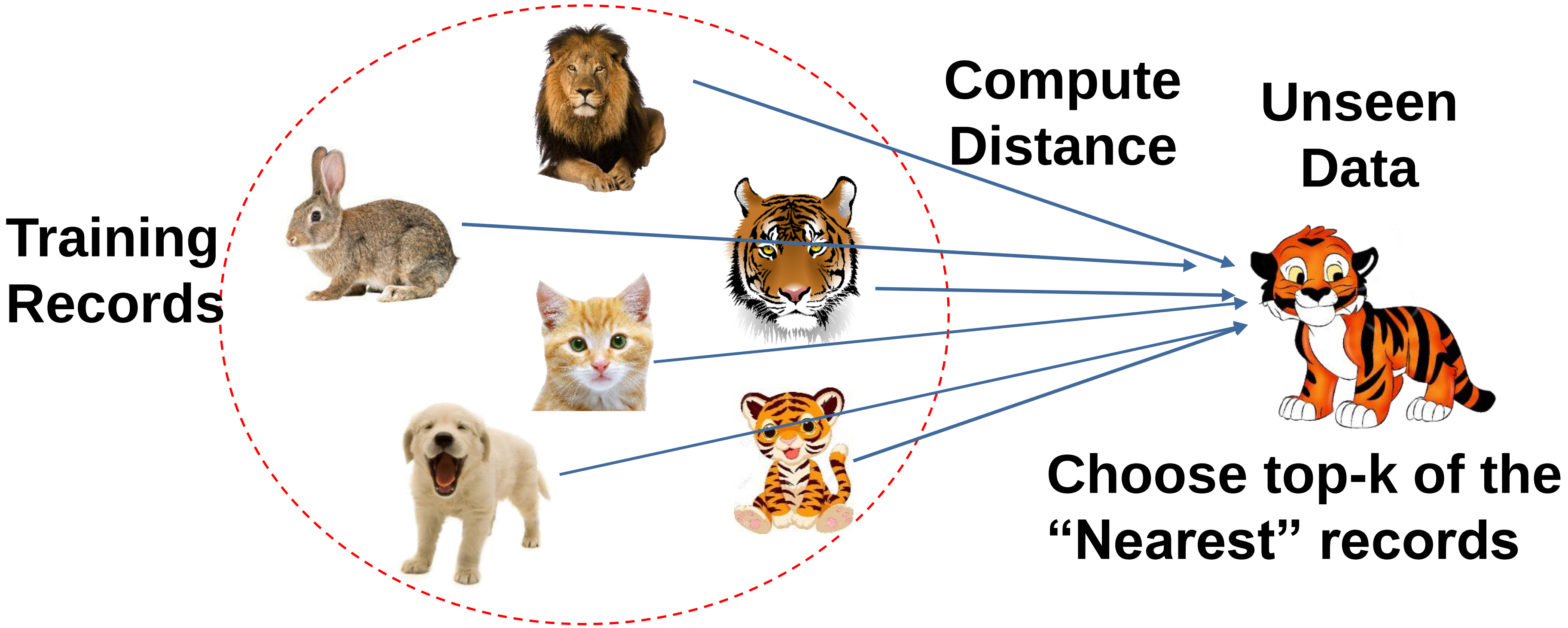
K-Nearest Neighbor (KNN) Classifier

- Problem: Identify which animal the given object it is
- Features: weights, age, gender, stripes, size, etc



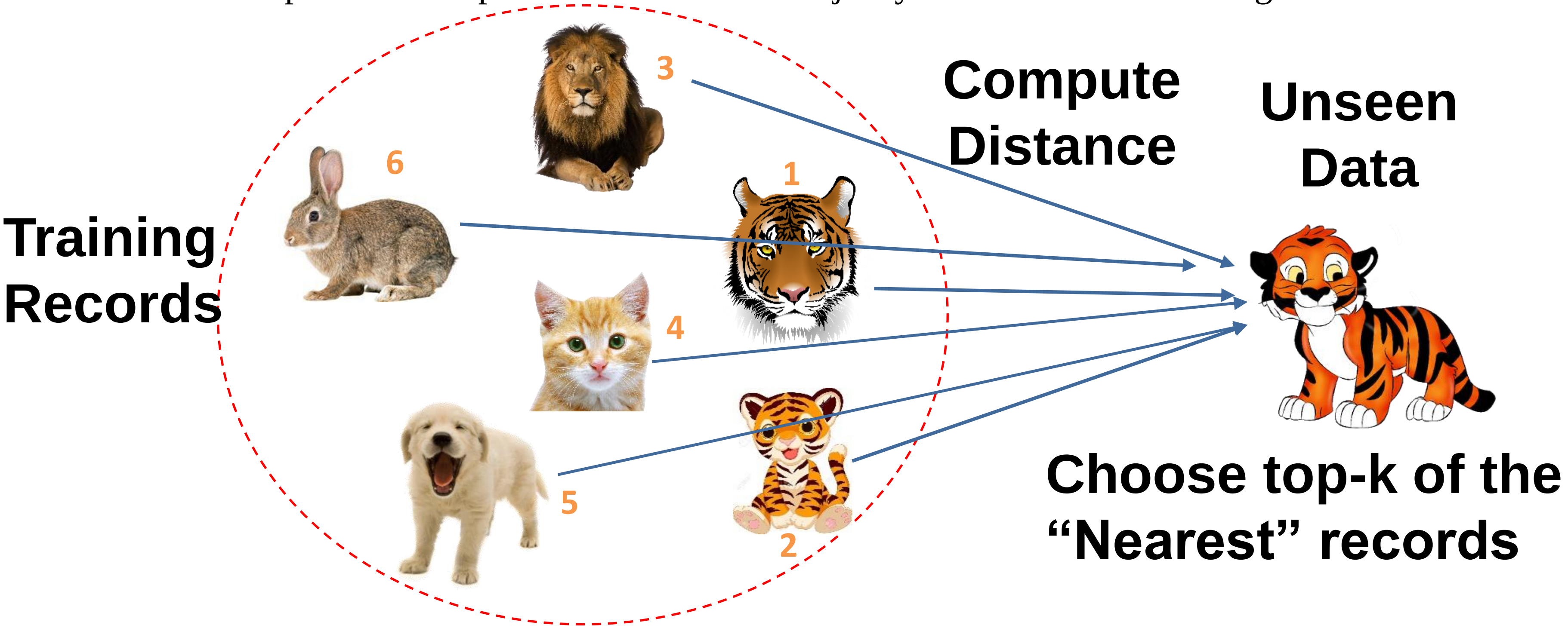
K-Nearest Neighbor (KNN) Classifier

- KNN classifier is a simple classification algorithm
- The idea behind is to classify new examples based on their similarity to or distance from examples we have seen before (in training set).



Build a KNN Classifier

- 1. Calculate distances between target and instances in train set
- 2. Identify the top-K nearest neighbor (choose an odd number for K!)
- 3. Predict labels and validate with truth
  - How to predict? The predicted class = the majority class label in those neighbors



For example, among top 3 picks (K = 3), 2/3 are tigers!!

Distance Measures

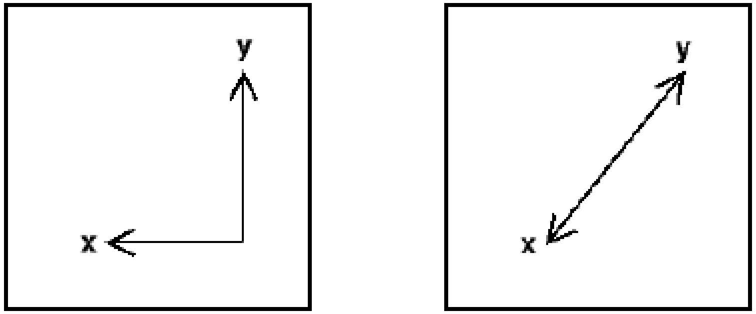
Assume there are *n* features, and two examples: *X* and *Y*.

- **Consider two vectors**
  - Rows in the data matrix $X = \langle x_1, x_2, \cdots, x_n \rangle$ $Y = \langle y_1, y_2, \cdots, y_n \rangle$
- **Common Distance Measures:**
  - Manhattan distance: (aggregation of two right-angle legs)

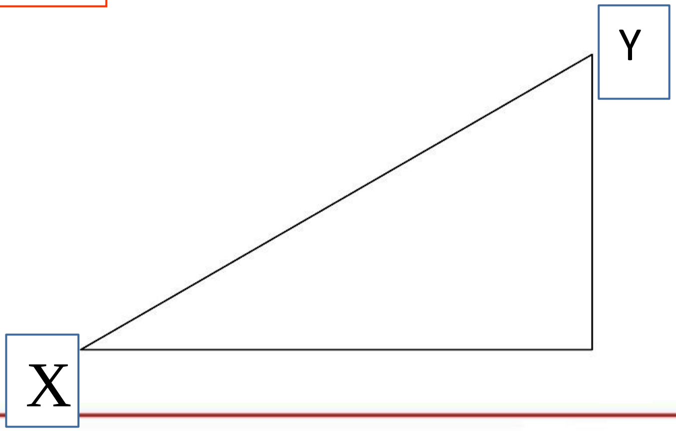
$$dist(X,Y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|$$

  - Euclidean distance: (length of hypotenuse)

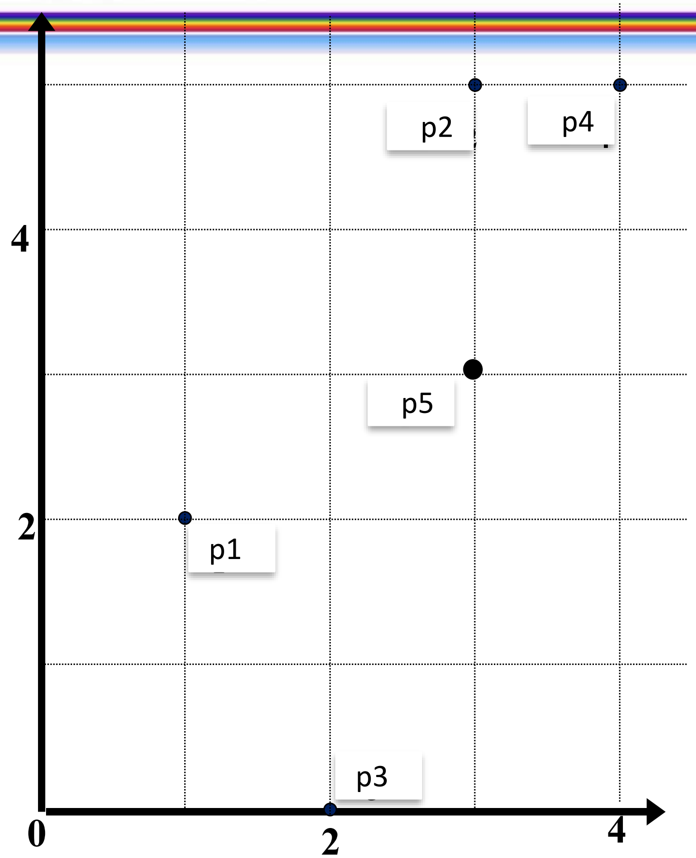
$$dist(X,Y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}$$



Manhattan Euclidean



Example: Distance Measures



$$dist(X,Y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}$$

Data Matrix

point	feature1	feature 2	class
p1	1	2	Y
p2	3	5	N
p3	2	0	Y
p4	4	5	N
p5	3	3	N

Distance Matrix (Euclidean)

	p1	***p2***	***p3***	***p4***	***p5***
p1	0				
p2	3.61	0			
p3	2.24	5.1	0		
p4	4.24	1	5.39	0	
p5	2.24	2	3.16	2.24	0

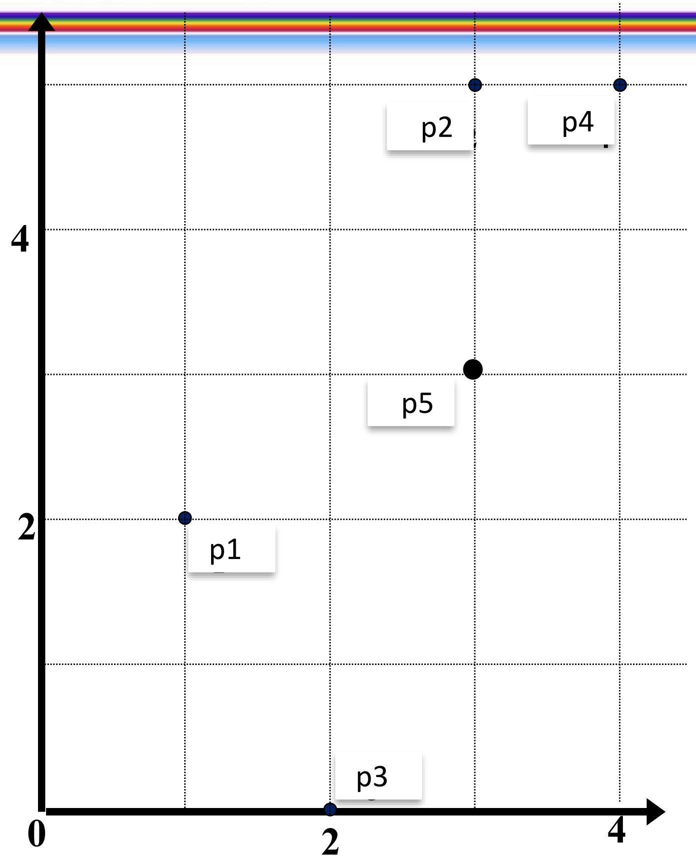
Set K = 3

p1' KNN = {*p3*, *p5*, *p2*}

p4' KNN = {*p2*, *p5*, *p1*}

Predict class for *p1* = N

Time for Practice!



Data Matrix

point	feature1	feature 2	class
p1	1	2	Y
p2	3	5	N
p3	2	0	Y
p4	4	5	N
p5	3	3	N

Distance Matrix (Manhattan)

	p1	***p2***	***p3***	***p4***	***p5***
p1	0				
p2		0			
p3			0		
p4				0	
p5					0

$$dist(X,Y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|$$

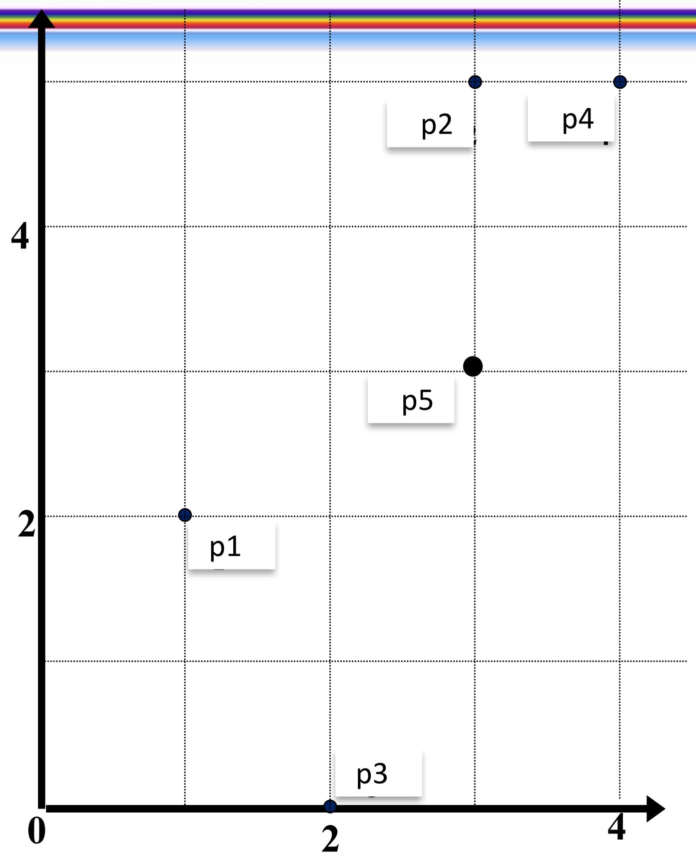
1.1) Set K = 3

p1' KNN = {*?, ?, ?*}

p4' KNN = {*?, ?, ?*}

1.2) Predict class for *p4* = ?

Answers!



Data Matrix

point	feature1	feature 2	class
p1	1	2	Y
p2	3	5	N
p3	2	0	Y
p4	4	5	N
p5	3	3	N

Distance Matrix (Manhattan)

	p1	*p2*	*p3*	*p4*	*p5*
p1	0				
p2	5	0			
p3	3	6	0		
p4	6	1	7	0	
p5	3	2	4	3	0

$$dist(X,Y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|$$

1.1) Set K = 3

p1' KNN = {*p3*, *p5*, *p2*}

p4' KNN = {*p2*, *p5*, *p1*}

1.2) Predict class for *p4* = N

Classification Algorithm:
K-Nearest Neighbor Classifier

More Questions

General Questions for Classifications

- What are the required data types by an algorithm
- Is there an overfitting problem?
- Is there a training-learning process?

General Questions for Classifications

- What are the required data types by an algorithm
- Is there an overfitting problem?
- Is there a training-learning process?

KNN: Features must be numerical

point	feature1	feature2	class
x1	1	2	Y
x2	3	5	N
x3	2	0	Y
x4	4	5	N
x5	3	3	N

Color	Weight (lbs)	Stripes	Tiger?
Orange	300	no	no
White	50	yes	no
Green	490	yes	yes
White	510	yes	yes
Orange	490	no	no

Answer: Convert a categorical feature to binary features

Color	Weight (lbs)	Stripes
Orange	300	no
White	50	yes
Green	490	yes
White	510	yes
Orange	490	no

Orange	White	Green	Weight (lbs)	Stripes
1	0	0	300	0
0	1	0	50	1
0	0	1	490	1
0	1	0	510	1
1	0	0	490	0

KNN: Features must be normalized

Feature normalization is used to convert values in a feature to the same scales with values in other features.

Orange	White	Green	Weight (lbs)	Stripes
1	0	0	300	0
0	1	0	50	1
0	0	1	490	1
0	1	0	510	1
1	0	0	490	0

Answer: Yes, normalization is required, otherwise, the distance calculation will be influenced by the larger features!!!!

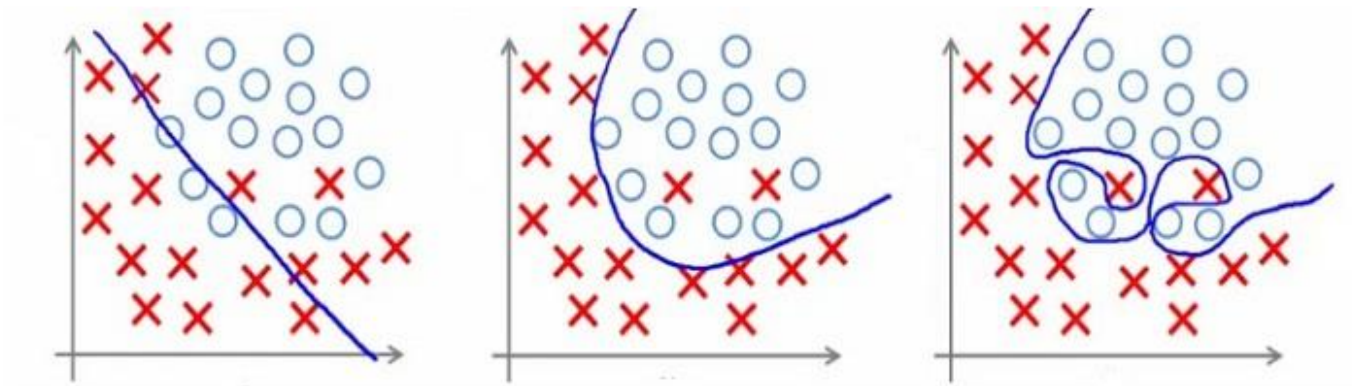
Min-max normalization: transformation from OldValue to NewValue

$$NewValue = NewMin + \frac{OldValue - OldMin}{OldMax - OldMin} \times (NewMax - NewMin)$$

General Questions for Classifications

- What are the required data types by an algorithm
- Is there an overfitting problem?
- Is there a training-learning process?

Overfitting Problem



Under-fitting

(too simple to explain the variance)

Appropriate-fitting

Over-fitting

(forcefitting -- too good to be true)

KNN: Overfitting Problem

- K value cannot be too small => overfitting!
  You make decisions based on a small neighborhood
  It is possible to have bias in the model!
- K value cannot be too large => underfitting!
  You make decisions based on a large neighborhood
- How to find the best K?
  - Try different K values in your experiments
    Do not always try 1, 3, 5, …, consider size of the data
  - Evaluate them in the correct strategy, and observe classification performance

General Questions for Classifications

- What are the required data types by an algorithm
- Is there an overfitting problem?
- Is there a training-learning process?

KNN: Learning Process?

- KNN is a lazy-learned. There are no learning process
- A learning process must have optimizations or loss functions
- In KNN, we do not have optimization objective and methods. => machine learning

Summary

- K-Nearest Neighbor (KNN) Classifier

A simple classifier, a lazy learner

1).Choose an odd number for K

2).Calculate distances between target and instances in training set

3).Pick the top KNN and assign the majority label as prediction

- Extended Problems in Classification Algorithms

Q1. required data types?

Q2. Is there an overfitting problem?

Q3. Is there a learning process?

Note: they are general concerns in classification, not only KNN.