Data Analytics

Yong Zheng

Illinois Institute of Technology

Chicago, IL, 60616, USA

School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY

Schedule

- Classification
- Classification by KNN
- Data Preprocessing

Schedule

- Classification
- Classification by KNN
- Data Preprocessing

Predictive Models We learnt

- Multiple Linear Prediction
  - One dependent variable, y → Numerical variable
  - Multiple independent variables, x → No limitations
- Linear Time-Series Model
  - One dependent variable, y → Numerical variable
  - Data with timestamp
  - AR, MA, ARMA, ARIMA
- Classification Model
  - One dependent variable, y → Nominal variable
  - Multiple independent variables, x → No limitations

Supervised v.s. Unsupervised Learning

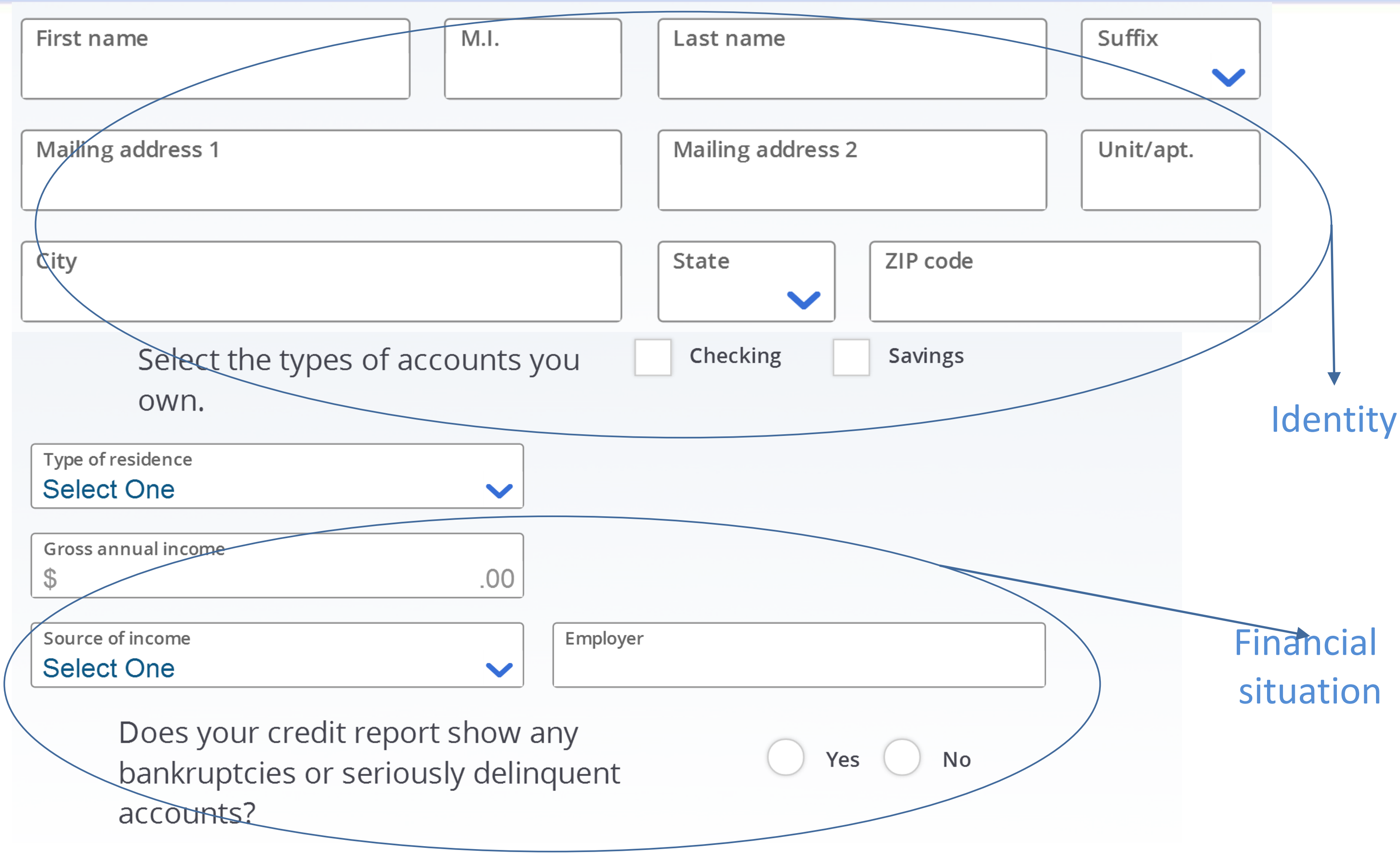
Machine Learning Algorithms *(sample)*

	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: 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
- Algorithms: How to perform classification tasks

Classification

- Classification Tasks
- Standard Classification Process
- Evaluation: How could we know it is good or bad
- 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?

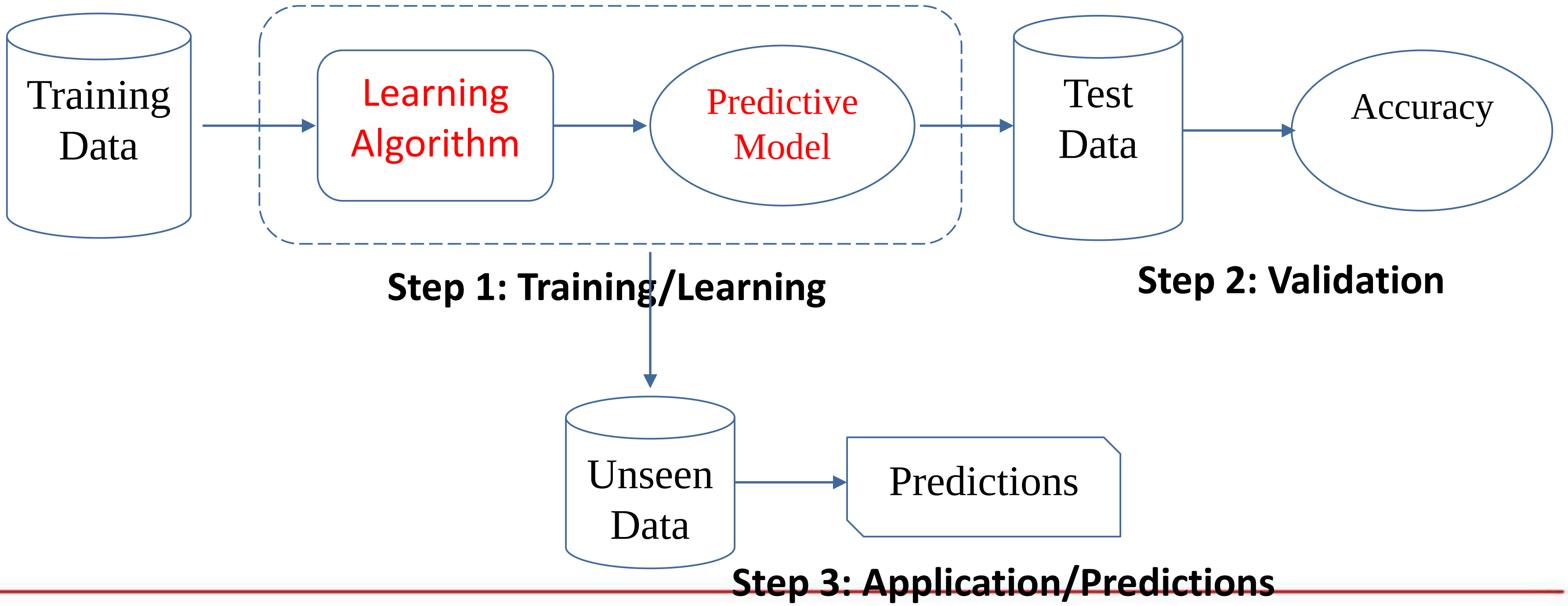
3). Multi-label Classification
Use appropriate words to describe it:
Red, Apple, Fruit, Tech, Mac, iPhone

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

Standard Process In Supervised Learning

- 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 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:

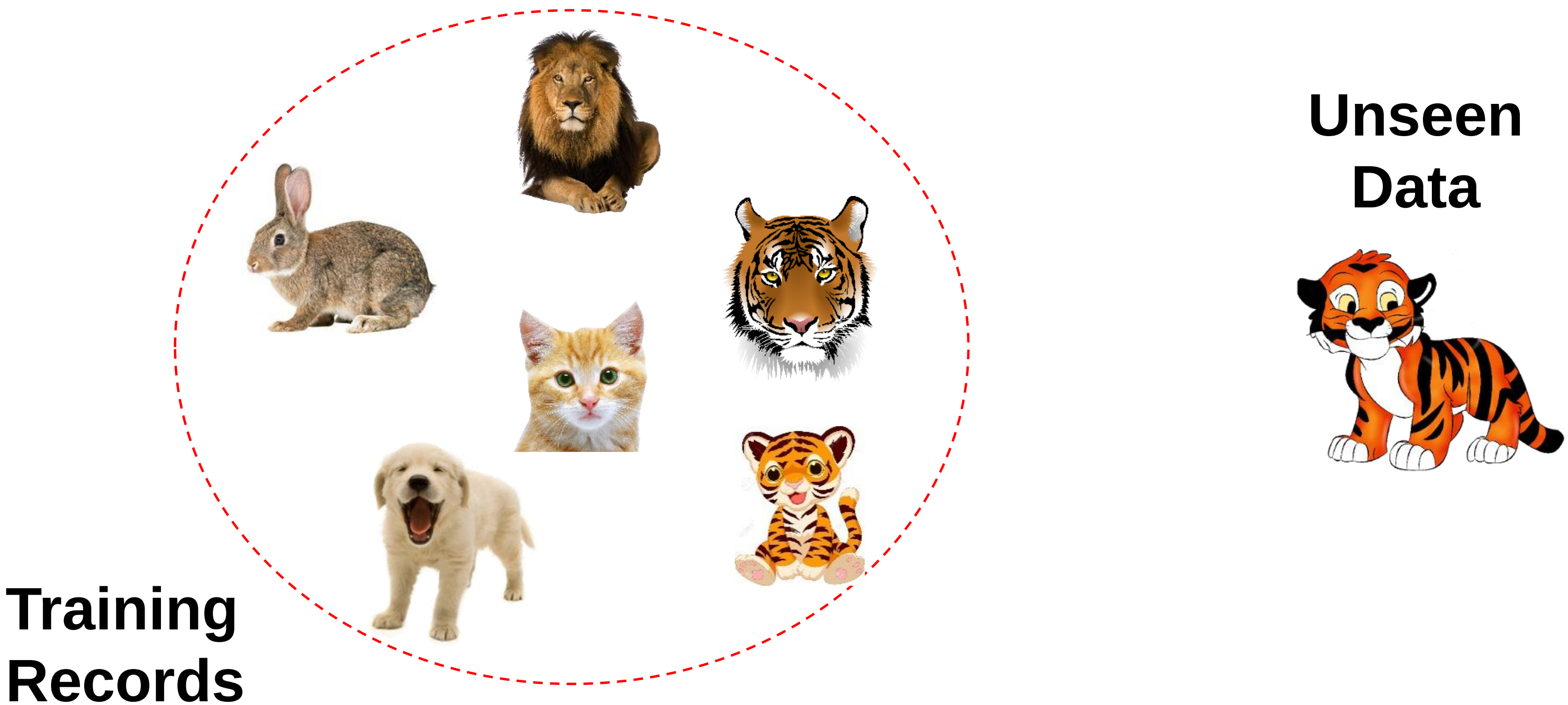
- K-nearest neighbor classifier
- Naïve Bayes classifier
- Decision tress
- Logistic regression
- Support Vector Machines
- Ensemble classifiers (e.g., random forest)
- …

Schedule

- Classification
- Classification by KNN
- Data Preprocessing

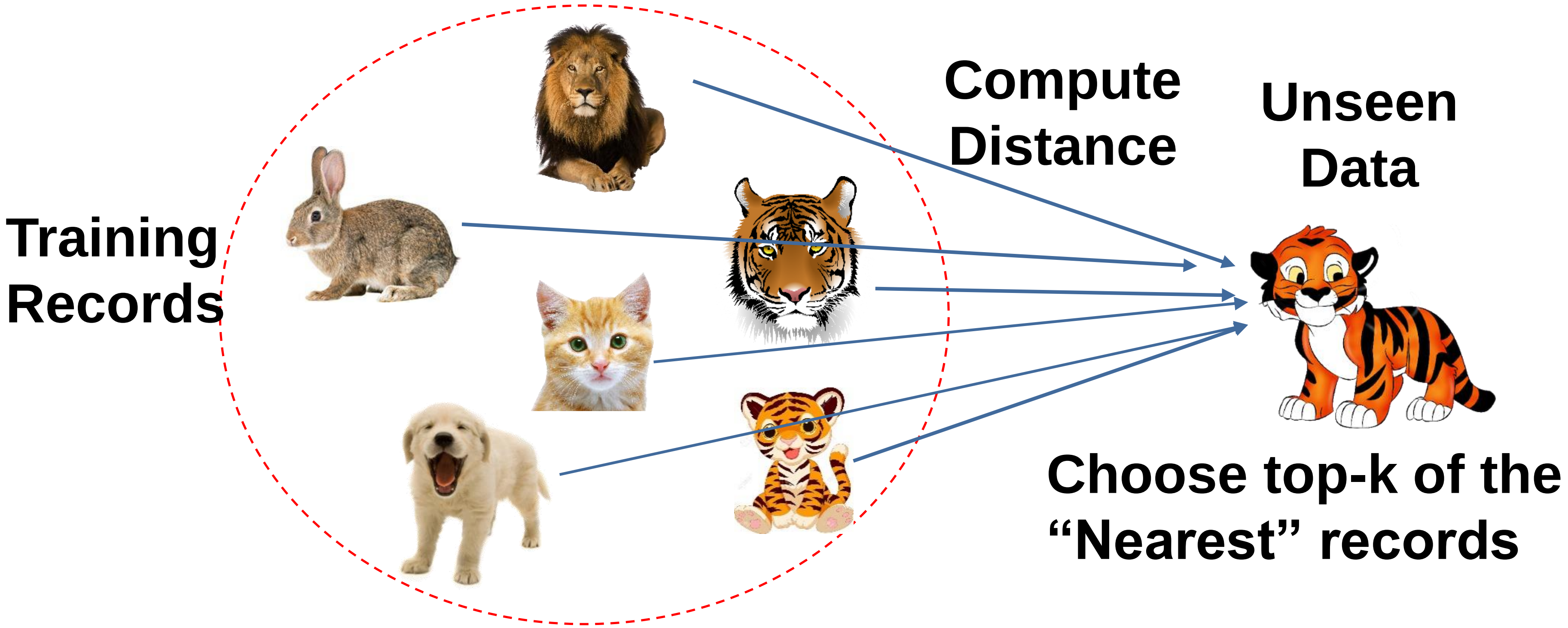
K-Nearest Neighbor (KNN) Classifier

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



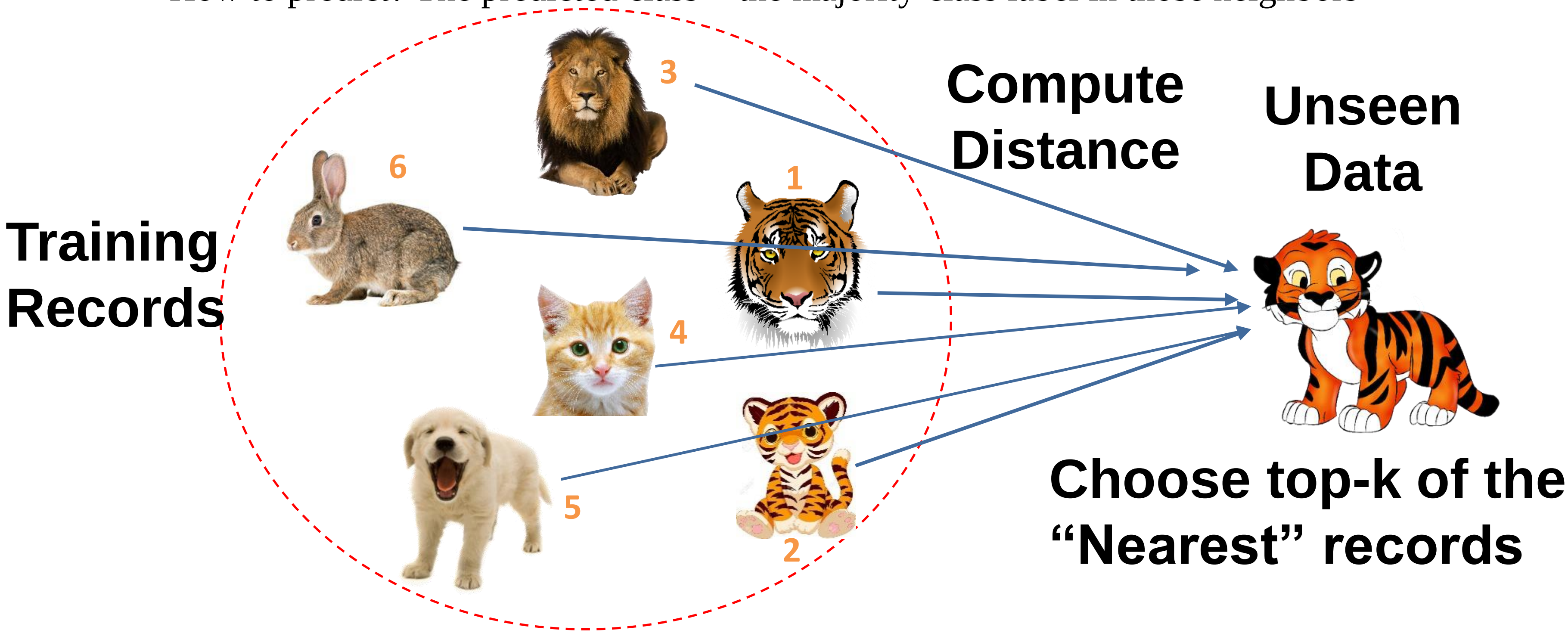
K-Nearest Neighbor (KNN) Classifier

- 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

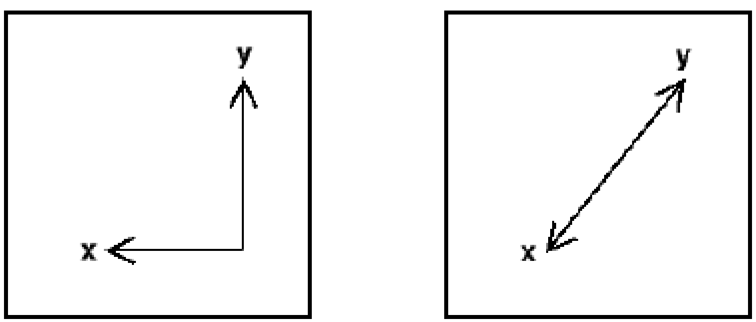
Assume there are *n* features, and two examples: *X* and *Y*.

- **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)

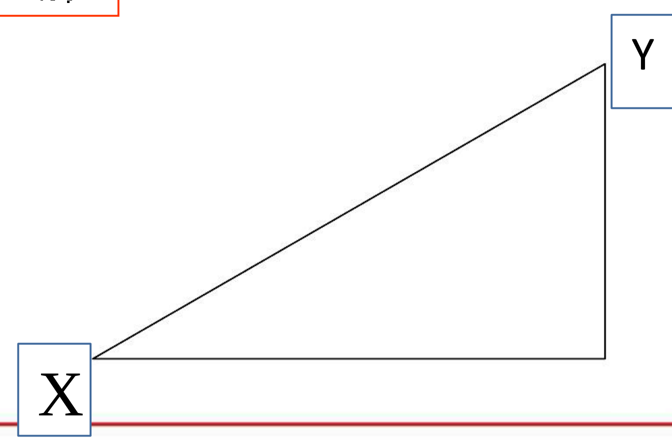
$$dist(X,Y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|$$

  - Euclidean distance: (length of hypotenuse)

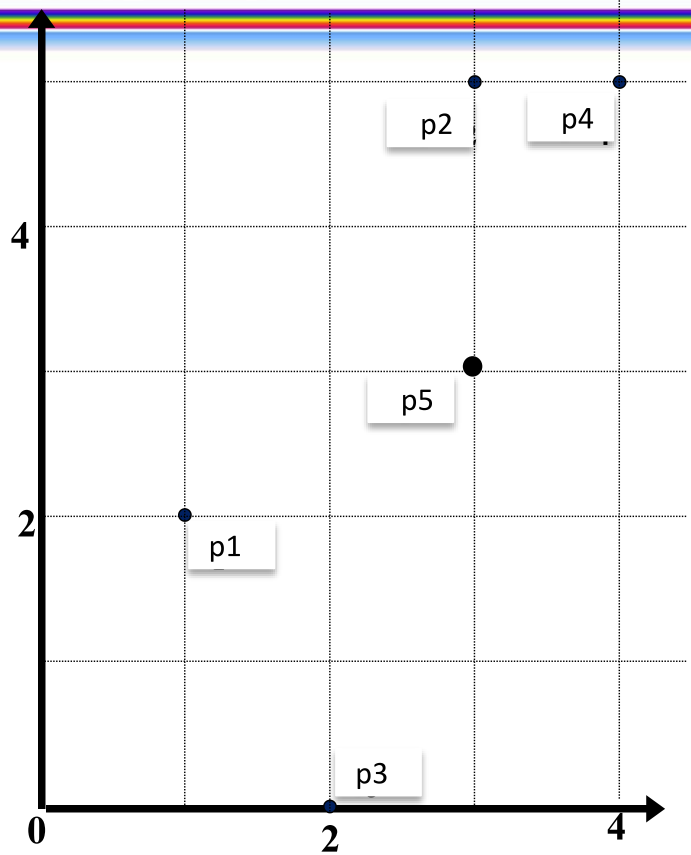
$$dist(X,Y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}$$



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	?

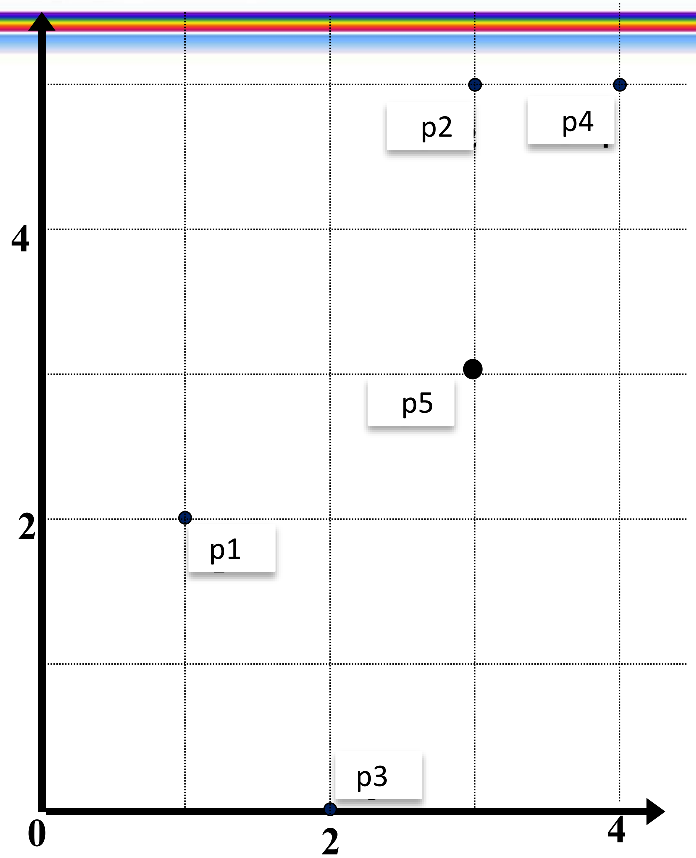
Distance Matrix (Euclidean)

	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

Predict label for p5

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	?

Distance Matrix (Manhattan)

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

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

Set K = 3

Predict label for p5

Classification Algorithm:
K-Nearest Neighbor Classifier

More Questions

Q1: How about categorical features?

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

Q2: Is feature normalization required?

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

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

Min-max normalization: transformation from OldValue to NewValue

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

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. Is it able to take categorical features? If Yes, how to treat them

Q2. Is normalization required?

Q3. How to alleviate overfitting problem?

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

R Practice

- German Credit Data

Download if from UCI ML Repository

http://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29

- Features and label

You need to read the page to understand the features and labels

Label = 1 (Good) or 2 (Bad) Credit

Features:

Existing checking account, credit history, purpose, credit amount, etc

You need to figure out feature types one by one

R Practice

❑ Load the data and take a look at it

```
setwd("C:/Users/cool/Desktop")
gc <- read.csv("germancredit.csv")
head (gc)
```

```
##   Default checkingstatus1 duration history purpose amount savings employ
## 1       0             A11        6     A34     A43   1169     A65    A75
## 2       1             A12       48     A32     A43   5951     A61    A73
## 3       0             A14       12     A34     A46   2096     A61    A74
## 4       0             A11       42     A32     A42   7882     A61    A74
## 5       1             A11       24     A33     A40   4870     A61    A73
## 6       0             A14       36     A32     A46   9055     A65    A73
##   installment status others residence property age otherplans housing
## 1           4    A93   A101         4     A121  67       A143    A152
## 2           2    A92   A101         2     A121  22       A143    A152
## 3           2    A93   A101         3     A121  49       A143    A152
## 4           2    A93   A103         4     A122  45       A143    A153
## 5           3    A93   A101         4     A124  53       A143    A153
## 6           2    A93   A101         4     A124  35       A143    A153
##   cards  job liable tele foreign
## 1     2 A173      1 A192    A201
## 2     1 A173      1 A191    A201
## 3     1 A172      2 A191    A201
## 4     1 A173      2 A191    A201
## 5     2 A173      2 A191    A201
## 6     1 A172      2 A192    A201
```

R Practice

- Data Preprocessing if necessary

Convert the dependent var to factor. Normalize the numeric variables

gc$Default <- factor(gc$Default) ➔ convert label to nominal data
num.vars <- sapply(gc, is.numeric) ➔ extract numerical variables
gc[num.vars] <- lapply(gc[num.vars], scale) ➔ normalize selected data, the function could be normalize or scale

Selecting only 3 numeric variables for this demostration, just to keep things simple

myvars <- c("Duration", "Amount", "Instalment")
gc.subset <- gc[myvars]

R Practice

- Evaluate model by hold-out evaluations

```
## predict on a test set of 100 observations. Rest to be used as train set.

set.seed(123)
test <- 1:100
train.gc <- gc.subset[-test,]
test.gc <- gc.subset[test,]

train.def <- gc$Default[-test]
test.def <- gc$Default[test]
```

R Practice

- User KNN to build the models and make predictions

```
library(class)

knn.1 <-  knn(train.gc, test.gc, train.def, k=1)
knn.5 <-  knn(train.gc, test.gc, train.def, k=5)
knn.15 <- knn(train.gc, test.gc, train.def, k=15)
```

R Practice

❑ Evaluation Based on the Accuracy

```
install.packages('Metrics', dependencies = TRUE)
 library(Metrics)
accuracy(actual data, predictions)
```

```
accuracy(test.def, knn.1)
accuracy(test.def, knn.5)
accuracy(test.def, knn.15)
```

```
> accuracy(test.def, knn.15)
[1] 0.9
> accuracy(test.def, knn.1)
[1] 0.72
> accuracy(test.def, knn.5)
[1] 0.78
> accuracy(test.def, knn.15)
[1] 0.9
```

R Practice

- How about N-fold cross-validation?

```
install.packages('caret', dependencies = TRUE)
library(caret)
x=gc.subset
y=gc$Default

model =
train(x,y,'knn',trControl=trainControl(method='cv',number=10),tuneGrid
= expand.grid(k = 1:10))
```

10-folds cross validation

Use K = 1, 2, 3, …, 10

The train function is very powerful. You can use several classification methods and evaluation methods. For more details https://machinelearningmastery.com/how-to-estimate-model-accuracy-in-r-using-the-caret-package/

R Practice

- How about N-fold cross-validation?

```
> model = train(x,y,'knn',trControl=trainControl(method='cv',number=10),tuneGrid   = expand.grid(k = 1:10))
> print(model)
k-Nearest Neighbors

1000 samples
   3 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 900, 900, 900, 900, 900, 900, ...
Resampling results across tuning parameters:

  k   Accuracy  Kappa
   1  0.594     0.04452968
   2  0.596     0.04595073
   3  0.638     0.07445211
   4  0.634     0.05021883
   5  0.648     0.04412389
   6  0.659     0.08326373
   7  0.659     0.04442904
   8  0.672     0.06202044
   9  0.681     0.06849165
  10  0.671     0.04058772
```

Schedule

- Classification
- Classification by KNN
- Data Preprocessing

Data Preprocessing by Using R

- Replace missing values by R
- Normalization by R
- Transformation: Numerical to Categorical
- Transformation: Categorical to Numerical

Data Preprocessing by Using R

- Replace missing values by R
- Normalization by R
- Transformation: Numerical to Categorical
- Transformation: Categorical to Numerical

Data Preprocessing by Using R

- Replace missing values by R

ave

Group Averages Over Level Combinations Of Factors

Subsets of `x[]` are averaged, where each subset consist of those observations with the same factor levels.

Keywords univar

Usage

```
ave(x, …, FUN = mean)
```

Arguments

x	A numeric.
...	Grouping variables, typically factors, all of the same `length` as `x`.
FUN	Function to apply for each factor level combination.

https://www.rdocumentation.org/packages/stats/versions/3.5.2/topics/ave

Data Preprocessing by Using R

- Replace missing values by R

ifelse

Conditional Element Selection

`ifelse` returns a value with the same shape as `test` which is filled with elements selected from either `yes` or `no` `TRUE` or `FALSE` .

Keywords programming, logic

Usage

```
ifelse(test, yes, no)
```

Arguments

test	an object which can be coerced to logical mode.
yes	return values for true elements of `test` .
no	return values for false elements of `test` .

https://www.rdocumentation.org/packages/base/versions/3.5.2/topics/ifelse

Data Preprocessing by Using R

- Replace missing values by R

```
##    Country Age Salary Purchased
## 1   France  44  72000        No
## 2    Spain  27  48000       Yes
## 3  Germany  30  54000        No
## 4    Spain  38  61000        No
## 5  Germany  40     NA       Yes
## 6   France  35  58000       Yes
## 7    Spain  NA  52000        No
## 8   France  48  79000       Yes
## 9  Germany  50  83000        No
## 10  France  37  67000       Yes
```

```
dataset$Age <- ifelse(is.na(dataset$Age),
                      ave(dataset$Age, FUN = function(x)
                        mean(x, na.rm = TRUE)),
                      dataset$Age)

dataset$Salary <- ifelse(is.na(dataset$Salary),
                      ave(dataset$Salary, FUN = function(x)
                        mean(x, na.rm = TRUE)),
                      dataset$Salary)
```

Data Preprocessing by Using R

- Replace missing values by R
- Normalization by R
- Transformation: Numerical to Categorical
- Transformation: Categorical to Numerical

Normalization by Using R

```
#extract numerical variables
num.vars<-sapply(data, is.numeric)
#normalize selected data using function scale
data[num.vars] <-lapply(data[num.vars], scale)

#min-max normalization to scale [0, 1]
data[num.vars] <-apply(data[num.vars], 2, FUN = function(x) (x - min(x))/(max(x)-min(x)))
```

Index for rows or columns
2 = apply function based on columns
1 = apply function based on rows

difference among apply(), lapply(), sapply(), tapply()?
https://www.guru99.com/r-apply-sapply-tapply.html

Data Preprocessing by Using R

- Replace missing values by R
- Normalization by R
- Transformation: Numerical to Categorical
- Transformation: Categorical to Numerical

Data PreProcessing

❑ Convert Numerical variable to Nominal variable in R

F1	F2	F3	F4	Class
C3	0	0	2	—
C2	1	0	5	+
C1	0	1	8	—
C2	1	1	16	—
C1	1	0	23	+
C3	0	1	11	+

Usually we use the cut function to create N groups

Data = cut(dataColumn, N)

```
> data=read.table("book1.csv", head=T, sep=',')
> data[,4]
[1]  2  5  8 16 23 11
> data[,4]=cut(data[,4], 3)
> data[,4]
[1] (1.98,9] (1.98,9] (1.98,9] (9,16]   (16,23]  (9,16]
Levels: (1.98,9] (9,16] (16,23]
> head(data)
  F1 F2 F3       F4 Class
1 C3  0  0 (1.98,9]     —
2 C2  1  0 (1.98,9]     +
3 C1  0  1 (1.98,9]     —
4 C2  1  1   (9,16]     —
5 C1  1  0  (16,23]     +
6 C3  0  1   (9,16]     +
```

Data Preprocessing by Using R

- Replace missing values by R
- Normalization by R
- Transformation: Numerical to Categorical
- Transformation: Categorical to Numerical

Data PreProcessing

- Convert Nominal Variable to Dummy variables in R

F1	F2	F3	F4	Class
C3	0	0	2	—
C2	1	0	5	+
C1	0	1	8	—
C2	1	1	16	—
C1	1	0	23	+
C3	0	1	11	+

```
install.packages("dummies")
library(dummies)
data=read.table("book1.csv", head=T, sep=',')
df=dummy.data.frame(data,names=c("F1"))
```

```
> df=dummy.data.frame(data,names=c("F1"))
>
> df
  F1C1 F1C2 F1C3 F2 F3 F4 Class
1    0    0    1  0  0  2     —
2    0    1    0  1  0  5     +
3    1    0    0  0  1  8     —
4    0    1    0  1  1 16     —
5    1    0    0  1  0 23     +
6    0    0    1  0  1 11     +
```

Note that it will create N dummy variables if there are N values in the nominal variable

Midterm Exam

- Time: Mar 26, 08:30 AM – 09:50 AM
- Location: SB 111
- Closed Note, Closed Book, Closed Devices
- You can bring a calculator. You can NOT share calculator with others.
- The questions in the exam will be the similar ones in your assignments; but you do not need to produce the R outputs, the outputs will be given in the exam papers.