CS4104 Machine Learning

K Nearest Neighbors Classifier (KNN)

Instance Based Learning

- First Example of Supervised Classification
- Rote-learner
  - Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly
- Nearest neighbor
  - Uses k “closest” points (nearest neighbors) for performing classification

Instance Based Learning

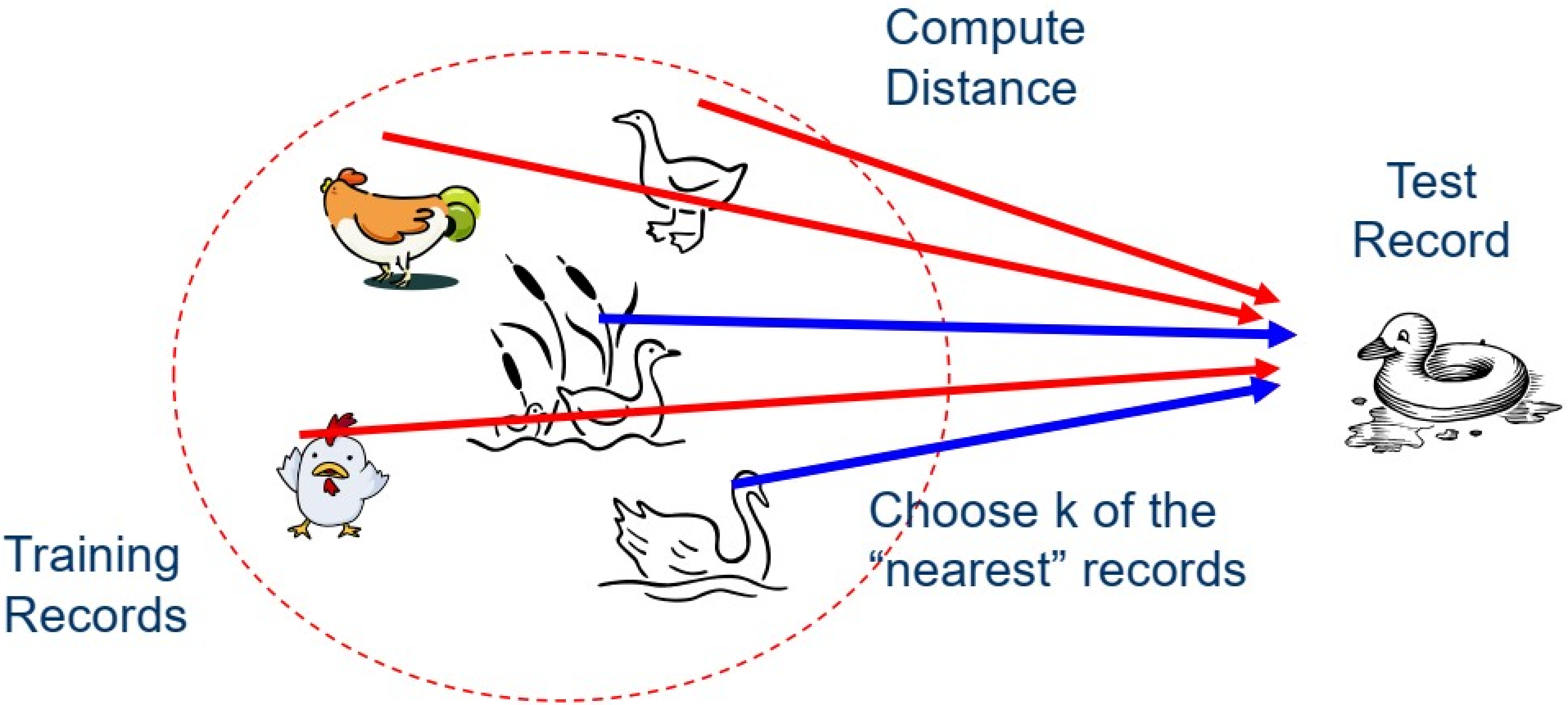
Labeled Data

Att1	Att2	Class
1	2	A
5	7	B
2	5	A
4	2	B

Unlabeled Data

Att1	Att2	Class
1	2	?
2	6	?
3	4	?

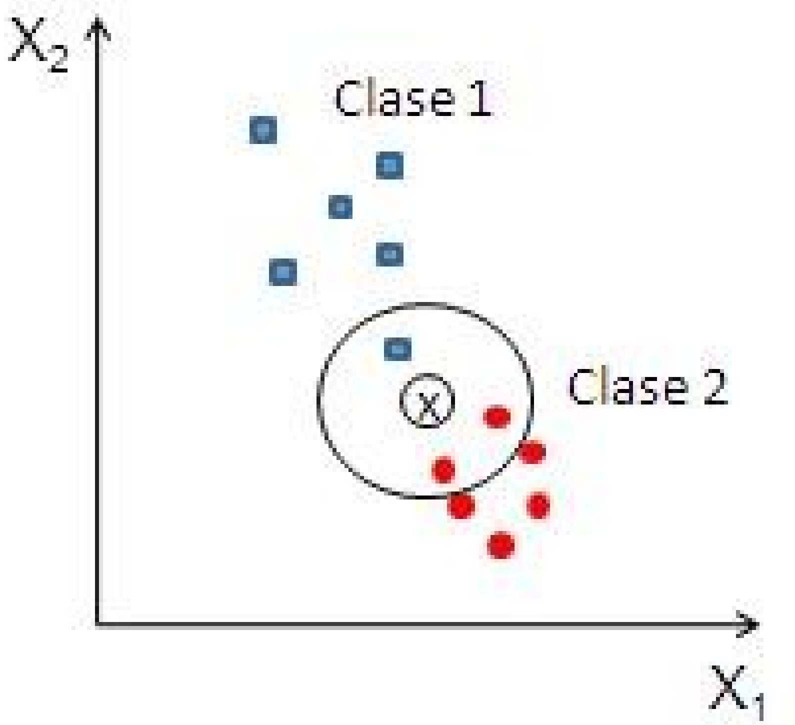
Nearest Neighbors



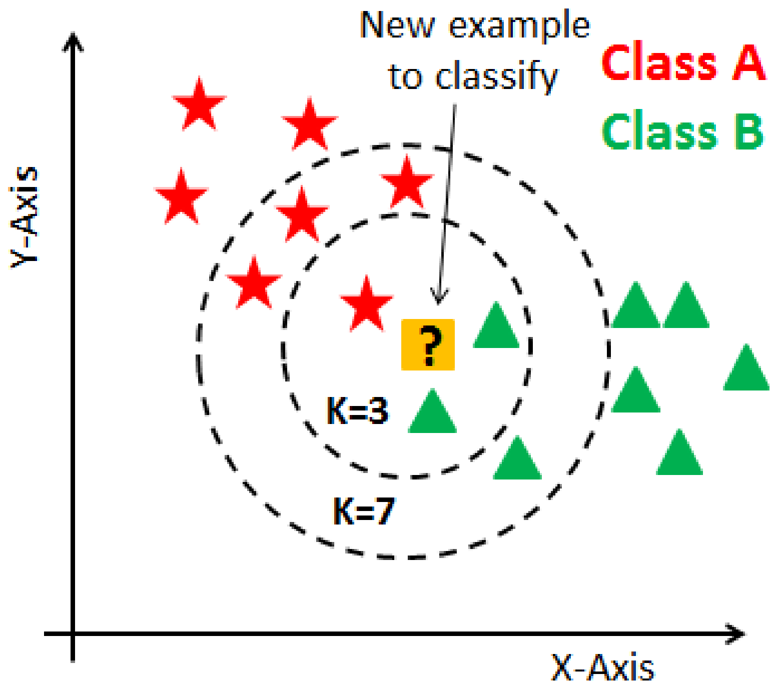
K Nearest Neighbors

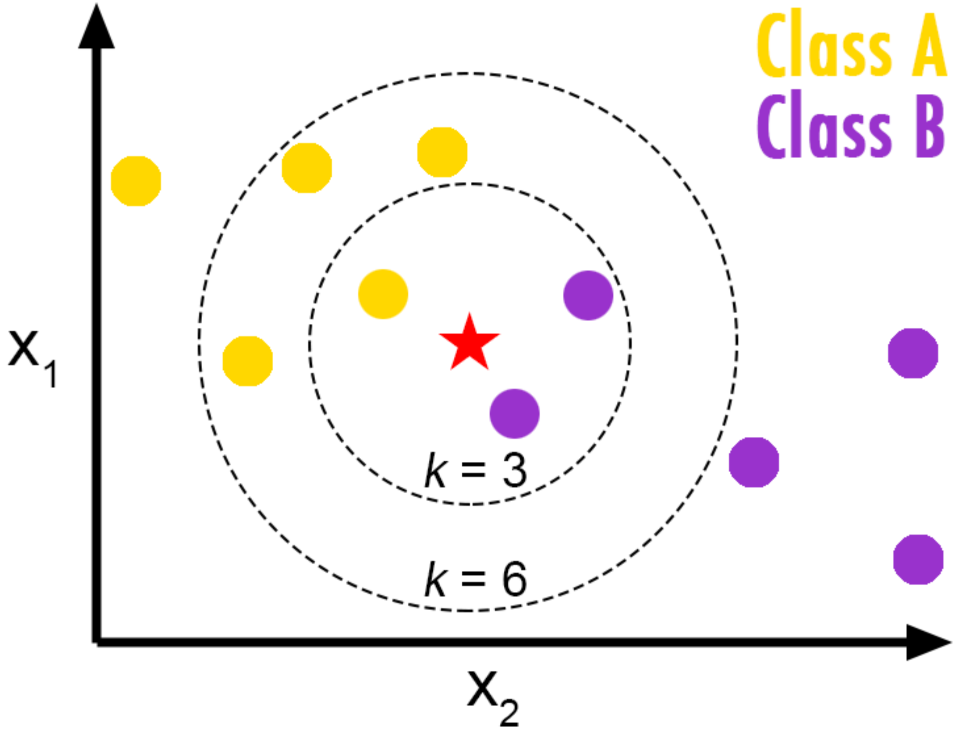
- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of k, the number of nearest neighbors to retrieve

- To classify an unknown record:
  1. Compute distance to other training records
  2. Identify k nearest neighbors
  3. Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)



K Nearest Neighbors (KNN)





zeshan.khan@nu.edu.pk

K Nearest Neighbors

1. Compute distance to other training records

2. Identify k nearest neighbors
3. Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Example

X1	X2	Class
1	3	B
2	4	B
3	2	A
5	4	A
2	5	?

- Assuming Distance as city block distance

Example

X1	X2	Class	Distance
1	3	B	|2-1|+|5-3|=3
2	4	B	|2-2|+|5-4|=1
3	2	A	|2-3|+|5-2|=4
5	4	A	|2-5|+|5-4|=4
2	5	?	

1. **Compute distance to other training records**
2. Identify k nearest neighbors
3. Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Example (k=1)

X1	X2	Class	Distance
1	3	B	|2-1|+|5-3|=3
2	4	B	|2-2|+|5-4|=1
3	2	A	|2-3|+|5-2|=4
5	4	A	|2-5|+|5-4|=4
2	5	?	

1. Compute distance to other training records
2. Identify k nearest neighbors
3. Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Example (k=2)

X1	X2	Class	Distance
1	3	B	|2-1|+|5-3|=3
2	4	B	|2-2|+|5-4|=1
3	2	A	|2-3|+|5-2|=4
5	4	A	|2-5|+|5-4|=4
2	5	?	

1. Compute distance to other training records
2. Identify k nearest neighbors
3. Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Example (k=1,2)

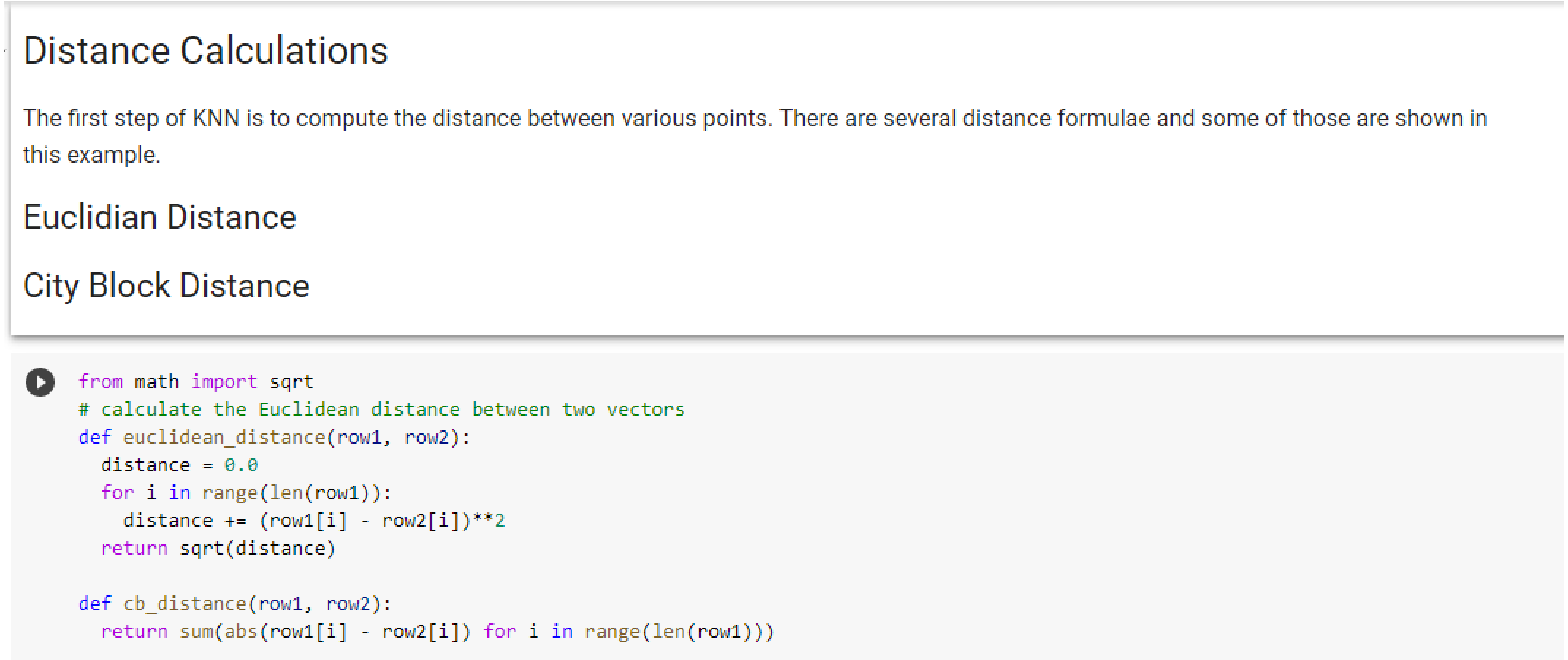
X1	X2	Class	Distance
1	3	B	|2-1|+|5-3|=3
2	4	B	|2-2|+|5-4|=1
3	2	A	|2-3|+|5-2|=4
5	4	A	|2-5|+|5-4|=4
2	5	B	

1. Compute distance to other training records
2. Identify k nearest neighbors
3. Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

KNN Code

- Constructor
- Train
- Test

Distance



Distance Calculations

The first step of KNN is to compute the distance between various points. There are several distance formulae and some of those are shown in this example.

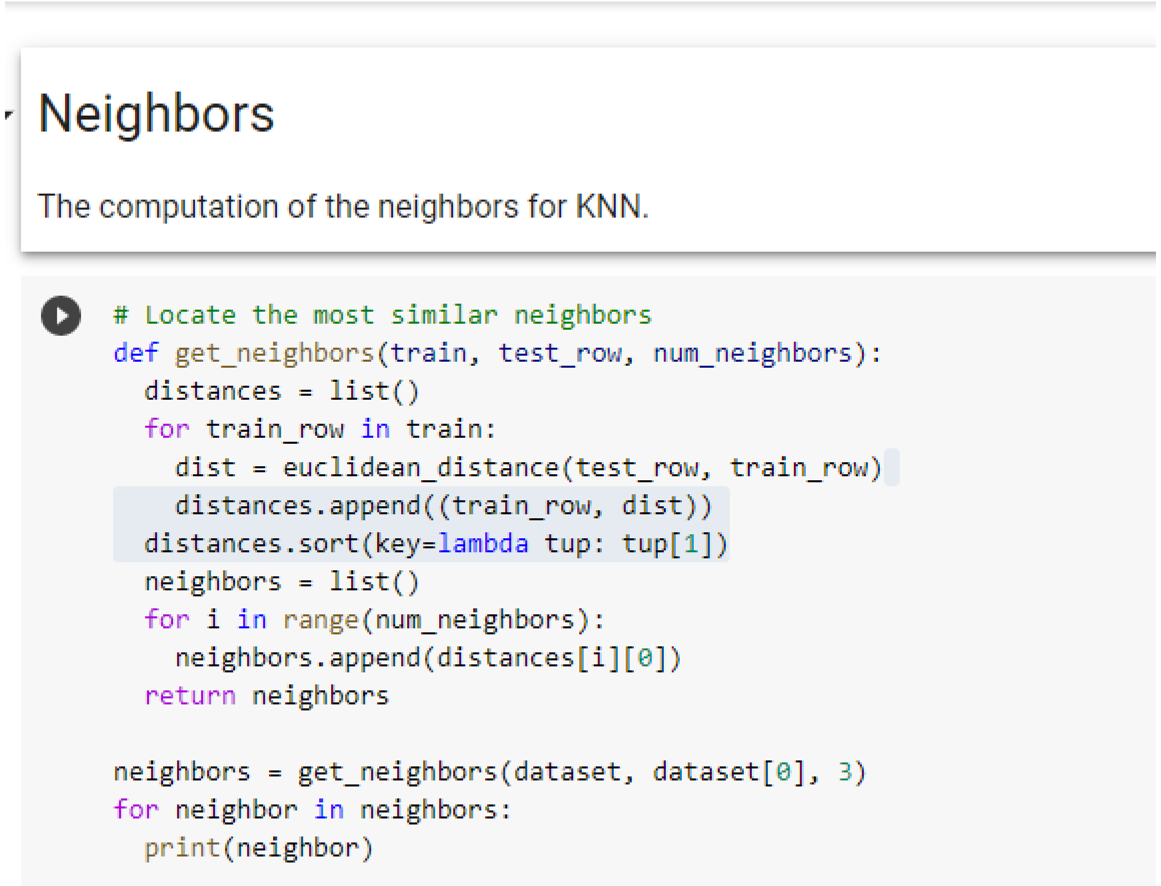
Euclidian Distance

City Block Distance

```
from math import sqrt
# calculate the Euclidean distance between two vectors
def euclidean_distance(row1, row2):
  distance = 0.0
  for i in range(len(row1)):
    distance += (row1[i] - row2[i])**2
  return sqrt(distance)

def cb_distance(row1, row2):
  return sum(abs(row1[i] - row2[i]) for i in range(len(row1)))
```

Neighbors



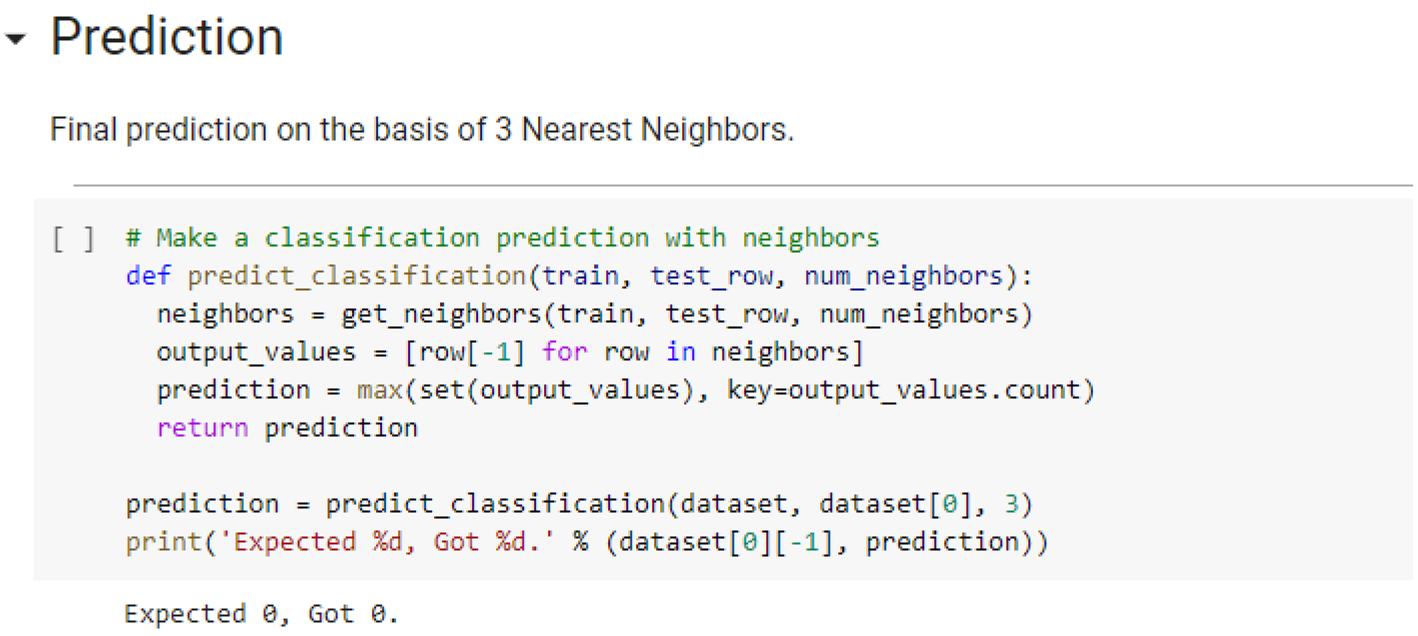
Neighbors

The computation of the neighbors for KNN.

```
# Locate the most similar neighbors
def get_neighbors(train, test_row, num_neighbors):
  distances = list()
  for train_row in train:
    dist = euclidean_distance(test_row, train_row)
    distances.append((train_row, dist))
  distances.sort(key=lambda tup: tup[1])
  neighbors = list()
  for i in range(num_neighbors):
    neighbors.append(distances[i][0])
  return neighbors

neighbors = get_neighbors(dataset, dataset[0], 3)
for neighbor in neighbors:
  print(neighbor)
```

Prediction



Prediction

Final prediction on the basis of 3 Nearest Neighbors.

```
[ ]  # Make a classification prediction with neighbors
     def predict_classification(train, test_row, num_neighbors):
       neighbors = get_neighbors(train, test_row, num_neighbors)
       output_values = [row[-1] for row in neighbors]
       prediction = max(set(output_values), key=output_values.count)
       return prediction

     prediction = predict_classification(dataset, dataset[0], 3)
     print('Expected %d, Got %d.' % (dataset[0][-1], prediction))
```

```
Expected 0, Got 0.
```

CS40104 Applied Machine Learning

Issues in KNN

Scale Effects

- Different features may have different measurement scales
  - E.g., patient weight in kg (range [50,200]) vs. blood protein values in ng/dL (range [-3,3])
- Consequences
  - Patient weight will have a much greater influence on the distance between samples
  - May bias the performance of the classifier

Standardization

- Transform raw feature values into z-scores

  - is the value for the i^{th} sample and j^{th} feature
  - is the average of all for feature *j*
  - is the standard deviation of all over all input samples

- Range and scale of z-scores should be similar (providing distributions of raw feature values are alike)

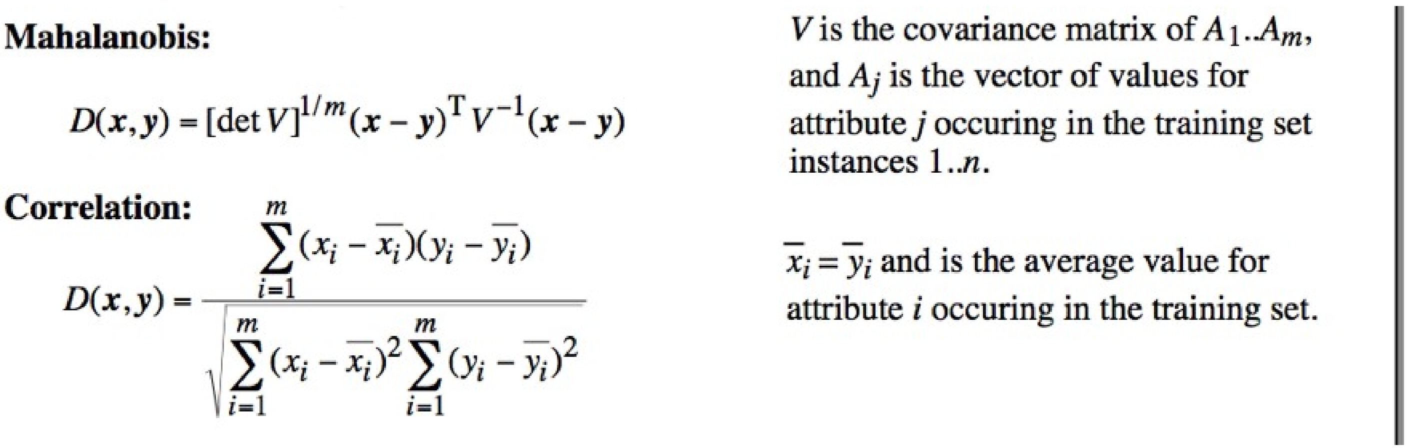
Distance Metrics

Distance Metrics…

- is an instance of a problem specific positive weight matrix

- is the sum of all values of attribute i in training set
- are the sums of all values in the vector x and y respectively.

Distance Metrics



Mahalanobis:

$$D(\boldsymbol{x},\boldsymbol{y}) = [\det V]^{1/m}(\boldsymbol{x}-\boldsymbol{y})^{\mathrm{T}}V^{-1}(\boldsymbol{x}-\boldsymbol{y})$$

V is the covariance matrix of $A_1..A_m$, and A_j is the vector of values for attribute j occuring in the training set instances $1..n$.

Correlation:

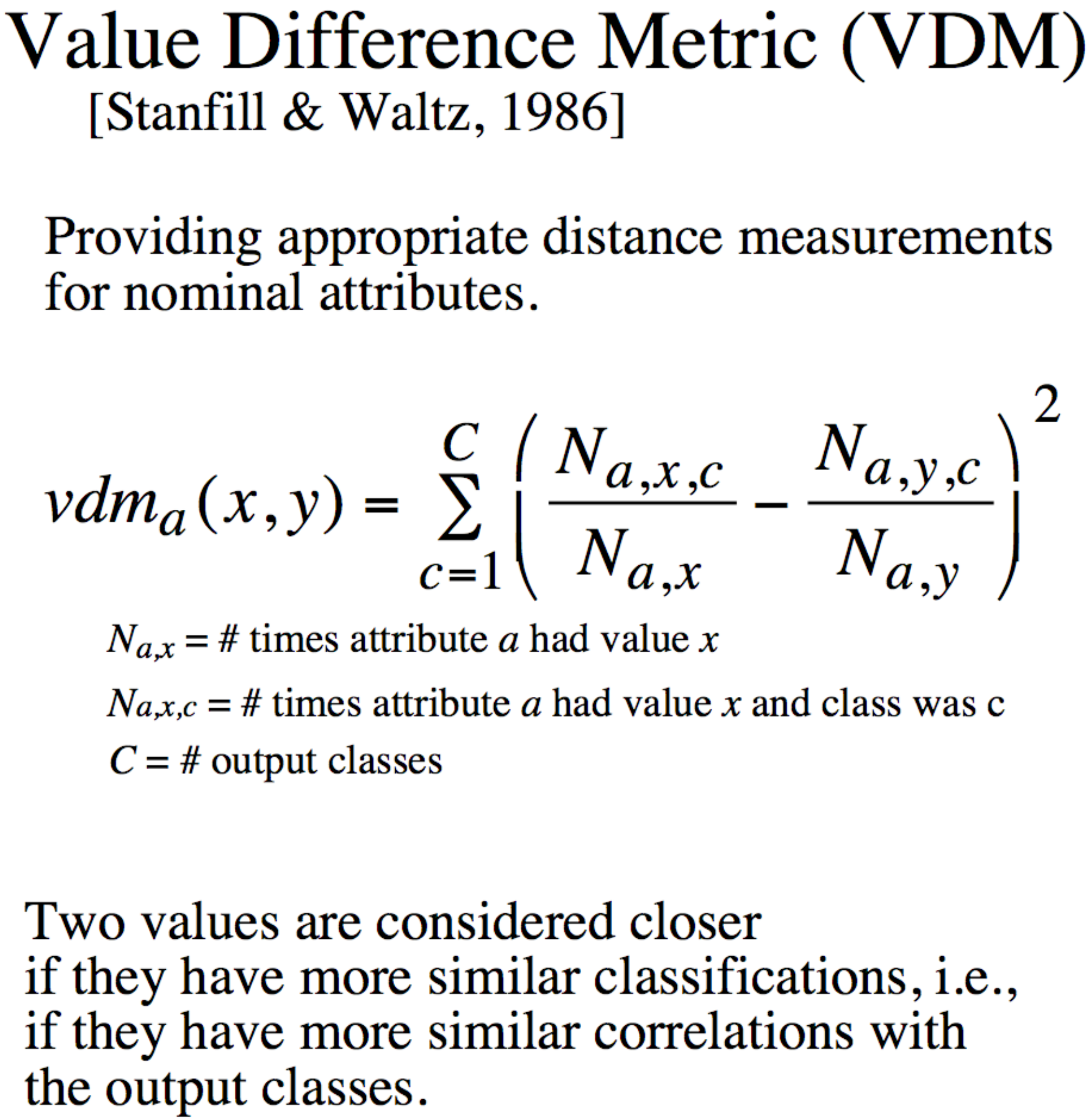
$$D(\boldsymbol{x},\boldsymbol{y}) = \frac{\sum_{i=1}^{m}(x_i-\overline{x_i})(y_i-\overline{y_i})}{\sqrt{\sum_{i=1}^{m}(x_i-\overline{x_i})^2\sum_{i=1}^{m}(y_i-\overline{y_i})^2}}$$

$\overline{x_i} = \overline{y_i}$ and is the average value for attribute i occuring in the training set.

Issues with Distance Metrics

- Most distance measures were designed for linear/real-valued attributes
- Two important questions in the context of machine learning:
  - How best to handle nominal attributes
  - What to do when attribute types are mixed

Distance for Nominal Attributes



Value Difference Metric (VDM)

[Stanfill & Waltz, 1986]

Providing appropriate distance measurements for nominal attributes.

$$vdm_a(x,y) = \sum_{c=1}^{C} \left(\frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right)^2$$

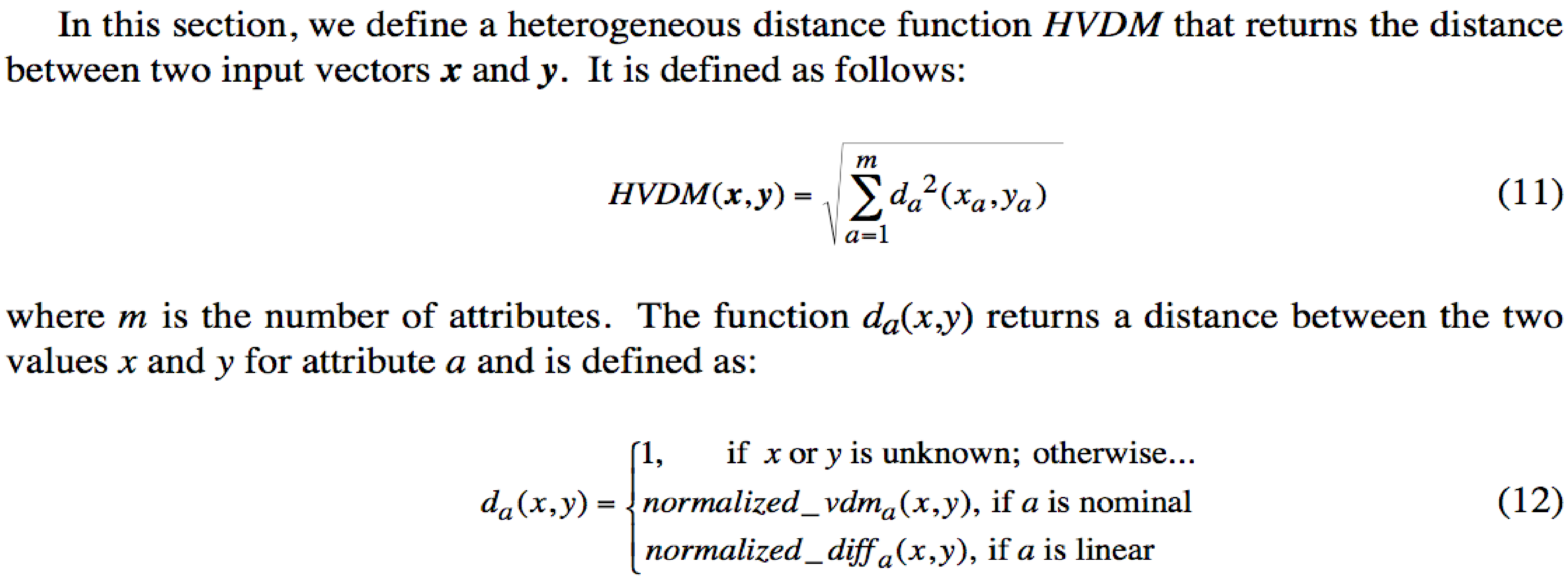
$N_{a,x}$ = # times attribute a had value x

$N_{a,x,c}$ = # times attribute a had value x and class was c

C = # output classes

Two values are considered closer
if they have more similar classifications, i.e.,
if they have more similar correlations with
the output classes.

Distance for Heterogeneous Data



In this section, we define a heterogeneous distance function *HVDM* that returns the distance between two input vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. It is defined as follows:

$$HVDM(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{a=1}^{m} d_a^{\,2}(x_a, y_a)} \qquad (11)$$

where m is the number of attributes. The function $d_a(x,y)$ returns a distance between the two values x and y for attribute a and is defined as:

$$d_a(x,y) = \begin{cases} 1, & \text{if } x \text{ or } y \text{ is unknown; otherwise...} \\ normalized\_vdm_a(x,y), & \text{if } a \text{ is nominal} \\ normalized\_diff_a(x,y), & \text{if } a \text{ is linear} \end{cases} \qquad (12)$$

Wilson, D. R. and Martinez, T. R., Improved Heterogeneous Distance Functions, Journal of Artificial Intelligence Research, vol. ***6****, no. 1, pp. 1-34, 1997*

Some Remarks

- k-NN works well on many practical problems and is fairly noise tolerant (depending on the value of k)
- k-NN is subject to the curse of dimensionality (i.e., presence of many irrelevant attributes)
- k-NN needs adequate distance measure
- k-NN relies on efficient indexing

Distance-weighted k-NN

- Replace

- by:

How is kNN Incremental?

- All training instances are stored
- Model consists of the set of training instances
- Adding a new training instance only affects the computation of neighbors, which is done at execution time (i.e., lazily)

  - Note that the storing of training instances is a violation of the strict definition of incremental learning.

Predicting Continuous Values

- Replace
- By: Replace
- Note: un-weighted corresponds to w_i=1 for all i

CS4104 Applied Machine Learning

Bayesian Classifier

Bayesian Theorem

- Conditional Probability

- Probability of Class C given Attribute A

- **Bayesian Theorem**

Example

- A doctor knows that polyps (P) causes GI-tract Cancer (C) 50% of the time
- Prior probability of any patient having Polyps (P) is 1/50,000
- Prior probability of any patient having GI-Track Cancer (C) is 1/20

-

Example 2: [Not Real Case]

- As per campus records, 20/400 students completed (C) their degree on time having short of attendance (A) in any subject.
- Every 10th student got shortage of attendance.
- 170 out of 340 students got completed their degree timely.
- Compute the probability of shortage of attendance for a student completed his degree timely.

-

Bayesian Classifier

- Consider each attribute and class label as random variables
- Given a record with attributes
  - Goal is to predict class C
  - Specifically, we want to find the value of C that maximizes
- Can we estimate directly from data?

Bayesian Classifier

- Approach
  - compute the posterior probability for all values of C using the Bayes theorem
  - Choose value of C that maximizes
  - Equivalent to choosing value of C that maximizes
- How to estimate

Naïve Bayes Classifier

- Assume independence among attributes Ai when class is given:

  - Can estimate for all and .
  - New point is classified to if is maximal.

Example

Dataset

Sr#	Refund	Status	Income	Cheat
1	Yes	1	50K	Yes
2	No	2	60K	Yes
3	Yes	1	10K	No
4	Yes	1	120K	No
5	Yes	2	101K	No
6	No	2	18K	Yes
7	No	1	87K	No
8	No	1	11K	No
9	Yes	2	20K	Yes
10	Yes	1	55K	?

Probabilities

- Discretize the range into bins
  - one ordinal attribute per bin
  - For income

Example

Dataset

Sr#	Refund	Status	Income	Cheat
1	Yes	1	B1	Yes
2	No	2	B2	Yes
3	Yes	1	B1	No
4	Yes	1	B2	No
5	Yes	2	B2	No
6	No	2	B1	Yes
7	No	1	B2	No
8	No	1	B1	No
9	Yes	2	B1	Yes
10	Yes	1	B2	?

Probabilities

- Discretize the range into bins
  - one ordinal attribute per bin
  - For income

Example

Dataset

Sr#	Refund	Status	Income	Cheat
1	Yes	1	B1	Yes
2	No	2	B2	Yes
3	Yes	1	B1	No
4	Yes	1	B2	No
5	Yes	2	B2	No
6	No	2	B1	Yes
7	No	1	B2	No
8	No	1	B1	No
9	Yes	2	B1	Yes
10	Yes	1	B2	?

Probabilities

Example

Dataset

Sr#	Refund	Status	Income	Cheat
1	Yes	1	B1	Yes
2	No	2	B2	Yes
3	Yes	1	B1	No
4	Yes	1	B2	No
5	Yes	2	B2	No
6	No	2	B1	Yes
7	No	1	B2	No
8	No	1	B1	No
9	Yes	2	B1	Yes
10	Yes	1	B2	?

Probabilities

Example

Probabilities

Test

Example

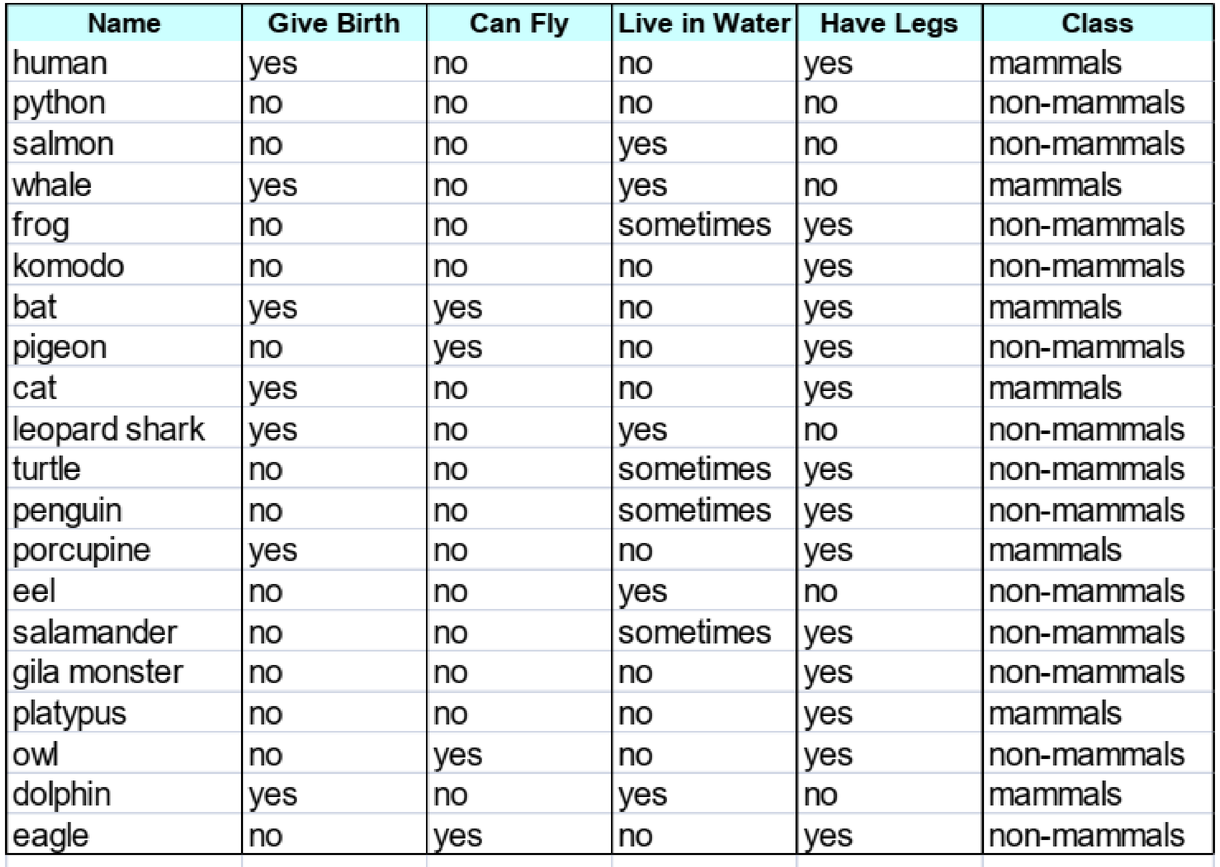
Probabilities

Test

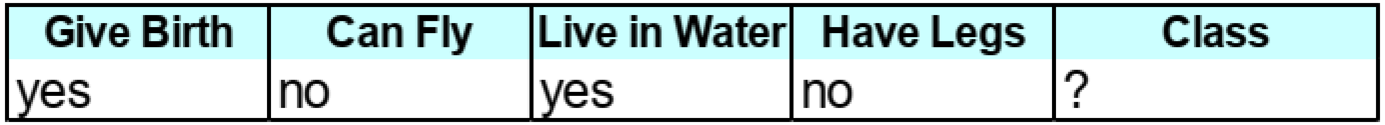
- Resultant class is No

Continues Variables Probabilities

Exercise

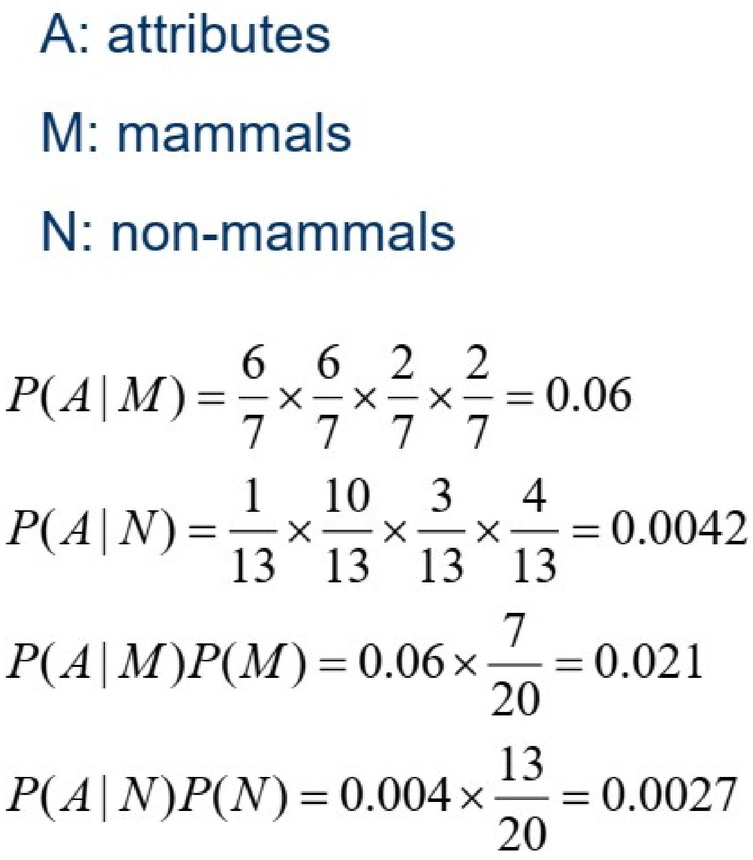


Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals



Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

Solution: Train



A: attributes

M: mammals

N: non-mammals

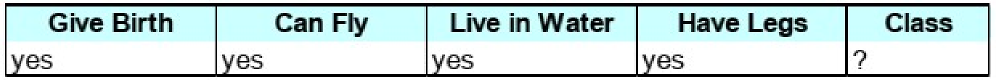
$$P(A|M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A|N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A|M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A|N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

Solution : Test



Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	yes	yes	yes	?

$$P(A|M) = \frac{6}{7} \times \frac{1}{7} \times \frac{2}{7} \times \frac{5}{7} = 0.025$$

$$P(A|N) = \frac{1}{13} \times \frac{3}{13} \times \frac{3}{13} \times \frac{9}{13} = 0.0028$$

$$P(A|M)P(M) = 0.025 \times \frac{7}{20} = 0.0088$$

$$P(A|N)P(N) = 0.004 \times \frac{13}{20} = 0.0018$$

P(A|M)P(M) > P(A|N)P(N)

=> Mammals

Naïve Bayes Analysis

- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes
  - Use other techniques such as Bayesian Belief Networks (BBN)