Introduction to Supervised Learning

Dr. Mahmoud N Mahmoud
mnmahmoud@ncat.edu

North Carolina A & T State University

August 28, 2020

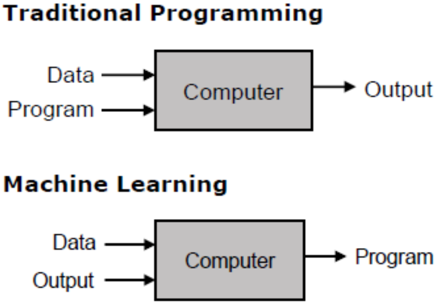
Talk Overview

1. Introduction to Machine Learning
2. ML Terminologies
3. ML Applications Examples
4. K-nearest Neighbours

Outline

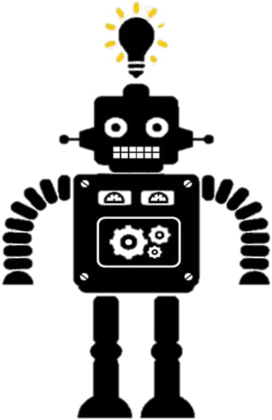
So What is Machine Learning?

- Automating automation
- Getting computers to program themselves
- Writing software is the bottleneck
- Let the data do the work instead!



WHAT IS MACHINE LEARNING?

Machine learning allows computers to learn and infer from data.



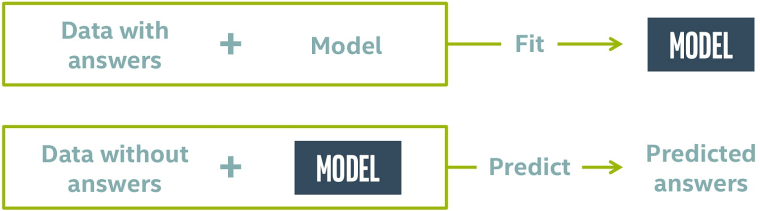
Machine Learning Problem Types

- **Based on Type of Data**
  - Supervised, Unsupervised, Semi supervised, Reinforcement Learning
- **Based on Type of Output**
  - Regression, Classification
- **Based on Type of Model**
  - Generative, Discriminative

Types of Learning based on Type of Data

- Supervised learning
  - Training data includes desired outputs.
  - Trying to learn a relation between input data and the output
- Unsupervised learning
  - Training data does not include desired outputs.
  - Trying to understand the data.
- Semi supervised learning
  - Training data includes a few desired outputs.
- Reinforcement learning
  - Rewards from sequence of actions.

SUPERVISED LEARNING OVERVIEW



Types of Learning based on Type of Output

Regression

A regression model predicts continuous values.

For example:

- What is the value of a house in California?
- What is the probability that a user will click on this ad?

Classification

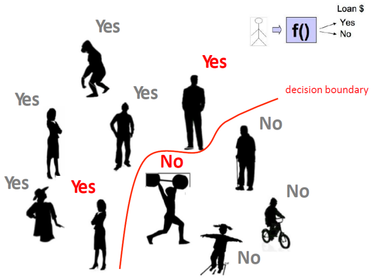
A classification model predicts discrete values.

For example:

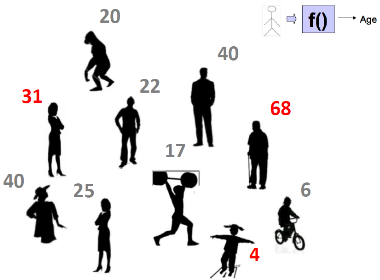
- Is a given email message spam or not spam?
- Is this an image of a dog, a cat, or a hamster?

Regression vs Classification

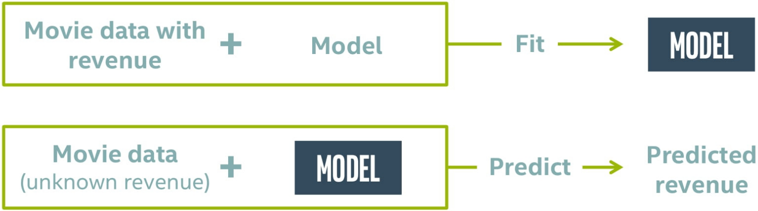
Classification (supervised learning)



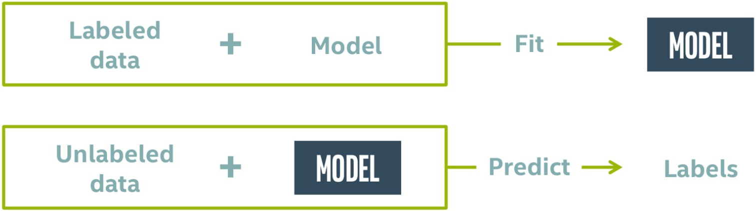
Regression (supervised learning)



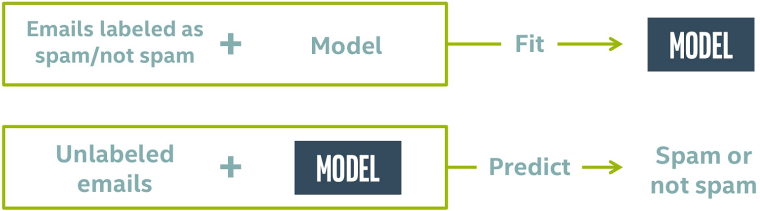
REGRESSION: NUMERICAL ANSWERS



CLASSIFICATION: CATEGORICAL ANSWERS



CLASSIFICATION: CATEGORICAL ANSWERS



Types of Learning based on Type of Model

Generative Model

A Generative nodel explicitly learns the actual distribution of each class.

Discriminative Model

A Discriminative model learns the decision boundary between the classes.

Generative Models

- Nave Bayes
- Hidden Markov Models
- Bayesian networks
- Markov random fields

Discriminative Models

- Logistic regression
- SVMs
- Traditional neural networks
- Nearest neighbor
- Conditional Random Fields (CRF)

Outline

ML Basic Terminologies

Many terminologies associated with ML. Will cover them in the following slides.

- Labels
- Features
- Examples
- Models

Label

- A label is the thing that we are predicting in classification or regression task. **Example:** Male, Female
- The label could also be the future price of wheat, the kind of animal shown in a picture, the meaning of an audio clip, or just about anything.
- Usually denoted with the variable y.

Features (1/2)

- A feature is an input variable (a.k.a attribute).
- A simple machine learning project might use a single feature, while a more sophisticated machine learning project could use millions of features.
- Usually denoted as:

$$x_1, x_2, \ldots, x_N$$

In the spam detector example, the features could include the following:

- words in the email text
- sender's address
- time of day the email was sent
- ...

Features (2/2)

Generally three types of attributes:

- **Categorical:** red, blue, brown, yellow
- **Ordinal:** poor, satisfactory, good, excellent
- **Numeric:** 3.14, 6E23, 0,

Categorical

- No natural ordering to categories
- Categories usually encoded as numbers

Ordinal

- There is a natural ordering to categories
- Encoded as numbers to preserve ordering

Numeric

- Integers or real numbers
- meaningful to add, mul tiply, compute

Notethat

The process of generating this features for our machine learning problem is called feature engineering.

Data samples (Examples)

Data sample / Example is a particular instance of data, **x**. (Note That. **x** is a vector of features)

We break examples into two categories:

- Labeled examples: (Used for prediction)
- Unlabeled examples: (Used for inference/testing)

Example

housingMedianAge (feature)	totalRooms (feature)	totalBedrooms (feature)	medianHouseValue (label)
15	5612	1283	66900
19	7650	1901	80100
17	720	174	85700
14	1501	337	73400
20	1454	326	65500

Model

A **model** defines a relationship between features and label.

Two phases of a model's life:

- **Training** means creating or learning the model. You show the model labeled examples and enable the model to gradually learn the relationships between features and label.
- **Testing/Inference** means applying the trained model to unlabeled examples. You use the trained model to make useful predictions (y').

Outline

ML Application 1: Credit Approval

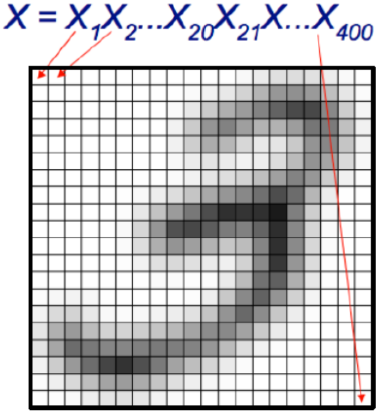
- **Numeric Features:**
  - loan amount (e. g. $1000)
  - Income (e. g. $65000)
- **Ordinal Features:**
  - savings: {none, <100, 100..500, 500..1000, >1000}
  - employed: {unemployed, <1yr, 1..4yrs, 4..7yrs, >7yrs}
- **Categorical Features:**
  - purpose: {car, appliance, repairs, education, business}
  - personal: {single, married, divorced, separated}
- **Labels (Categorical):**
  - Approve credit application
  - Disapprove credit application

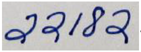
Easy feature engineering process.

ML Application 2: Handwritten Digits Recognation

Represent each pixel as a separate attribute either Categorical **OR** Ordinal:

- **Categorical Features:**
  - (white) or (black) based on a threshold
- **Ordinal Features:**
  - Degree of "blackness" of a pixel
- **Labels (Categorical):** {0,1,2,3,4,5,6,7,8,9}



What if we are dealing with paper like this  ?

Isolate each digit, rescale, de-slant, ..

Hard feature engineering process.

MACHINE LEARNING VOCABULARY

- **Target: predicted category or value of the data**
  (column to predict)

MACHINE LEARNING VOCABULARY

Sepal length	Sepal width	Petal length	Petal width	Species
6.7	3.0	5.2	2.3	Virginica
6.4	2.8	5.6	2.1	Virginica
4.6	3.4	1.4	0.3	Setosa
6.9	3.1	4.9	1.5	Versicolor
4.4	2.9	1.4	0.2	Setosa
4.8	3.0	1.4	0.1	Setosa
5.9	3.0	5.1	1.8	Virginica
5.4	3.9	1.3	0.4	Setosa
4.9	3.0	1.4	0.2	Setosa
5.4	3.4	1.7	0.2	Setosa

MACHINE LEARNING VOCABULARY

Sepal length	Sepal width	Petal length	Petal width	Species
6.7	3.0	5.2	2.3	Virginica
6.4	2.8	5.6	2.1	Virginica
4.6	3.4	1.4	0.3	Setosa
6.9	3.1	4.9	1.5	Versicolor
4.4	2.9	1.4	0.2	Setosa
4.8	3.0	1.4	0.1	Setosa
5.9	3.0	5.1	1.8	Virginica
5.4	3.9	1.3	0.4	Setosa
4.9	3.0	1.4	0.2	Setosa
5.4	3.4	1.7	0.2	Setosa

Target

MACHINE LEARNING VOCABULARY

- **Target: predicted category or value of the data**
  (column to predict)
- **Features: properties of the data used for prediction**
  (non-target columns)

MACHINE LEARNING VOCABULARY

Features

Sepal length	Sepal width	Petal length	Petal width	Species
6.7	3.0	5.2	2.3	Virginica
6.4	2.8	5.6	2.1	Virginica
4.6	3.4	1.4	0.3	Setosa
6.9	3.1	4.9	1.5	Versicolor
4.4	2.9	1.4	0.2	Setosa
4.8	3.0	1.4	0.1	Setosa
5.9	3.0	5.1	1.8	Virginica
5.4	3.9	1.3	0.4	Setosa
4.9	3.0	1.4	0.2	Setosa
5.4	3.4	1.7	0.2	Setosa

MACHINE LEARNING VOCABULARY

- **Target: predicted category or value of the data**
  (column to predict)
- **Features: properties of the data used for prediction**
  (non-target columns)
- **Example: a single data point within the data**
  (one row)

MACHINE LEARNING VOCABULARY

Examples →

Sepal length	Sepal width	Petal length	Petal width	Species
6.7	3.0	5.2	2.3	Virginica
6.4	2.8	5.6	2.1	Virginica
4.6	3.4	1.4	0.3	Setosa
6.9	3.1	4.9	1.5	Versicolor
4.4	2.9	1.4	0.2	Setosa
4.8	3.0	1.4	0.1	Setosa
5.9	3.0	5.1	1.8	Virginica
5.4	3.9	1.3	0.4	Setosa
4.9	3.0	1.4	0.2	Setosa
5.4	3.4	1.7	0.2	Setosa

MACHINE LEARNING VOCABULARY

- **Target: predicted category or value of the data**
  (column to predict)
- **Features: properties of the data used for prediction**
  (non-target columns)
- **Example: a single data point within the data**
  (one row)
- **Label: the target value for a single data point**

MACHINE LEARNING VOCABULARY

Sepal length	Sepal width	Petal length	Petal width	Species
6.7	3.0	5.2	2.3	Virginica
6.4	2.8	5.6	2.1	Virginica
4.6	3.4	1.4	0.3	Setosa
6.9	3.1	4.9	1.5	Versicolor
4.4	2.9	1.4	0.2	Setosa
4.8	3.0	1.4	0.1	Setosa
5.9	3.0	5.1	1.8	Virginica
5.4	3.9	1.3	0.4	Setosa
4.9	3.0	1.4	0.2	Setosa
5.4	3.4	1.7	0.2	Setosa

Label

Outline

WHAT IS CLASSIFICATION?

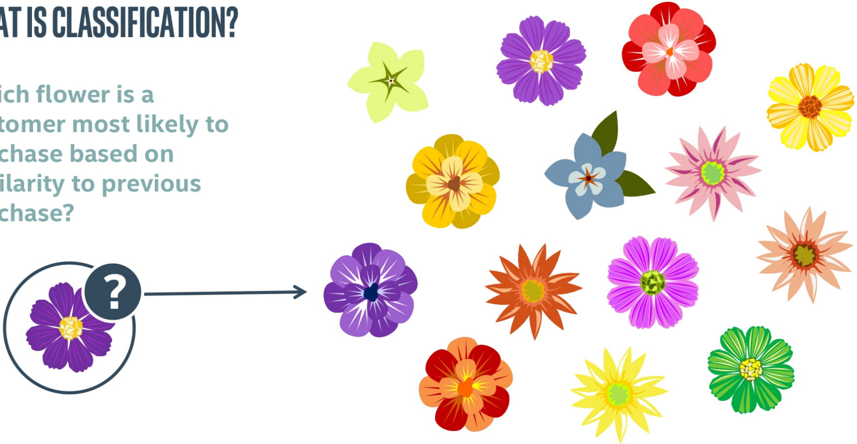
A flower shop wants to guess a customer's purchase from similarity to most recent purchase.

WHAT IS CLASSIFICATION?

Which flower is a customer most likely to purchase based on similarity to previous purchase?

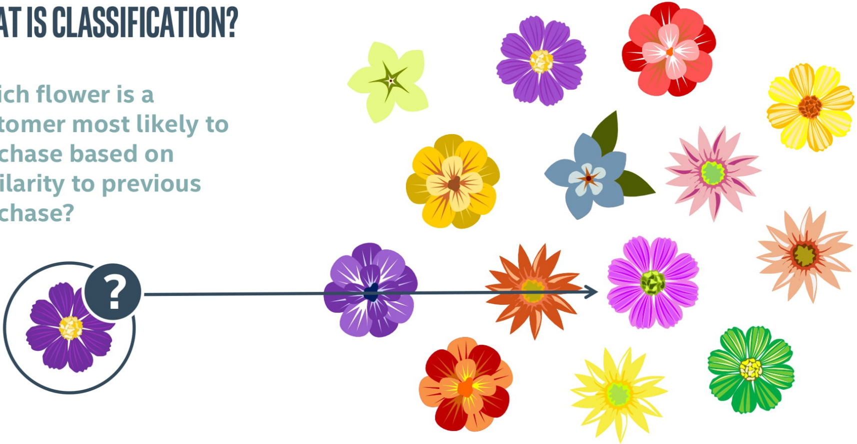
WHAT IS CLASSIFICATION?

Which flower is a customer most likely to purchase based on similarity to previous purchase?



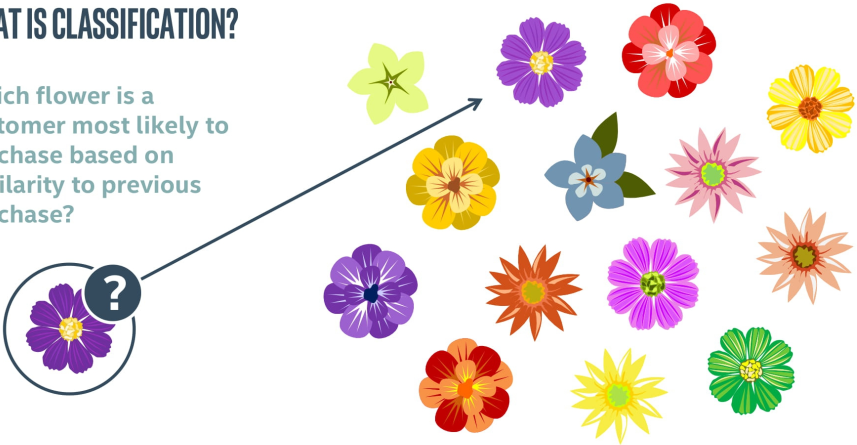
WHAT IS CLASSIFICATION?

Which flower is a customer most likely to purchase based on similarity to previous purchase?



WHAT IS CLASSIFICATION?

Which flower is a customer most likely to purchase based on similarity to previous purchase?



WHAT IS NEEDED FOR CLASSIFICATION?

- **Model data with:**
  - Features that can be quantitated

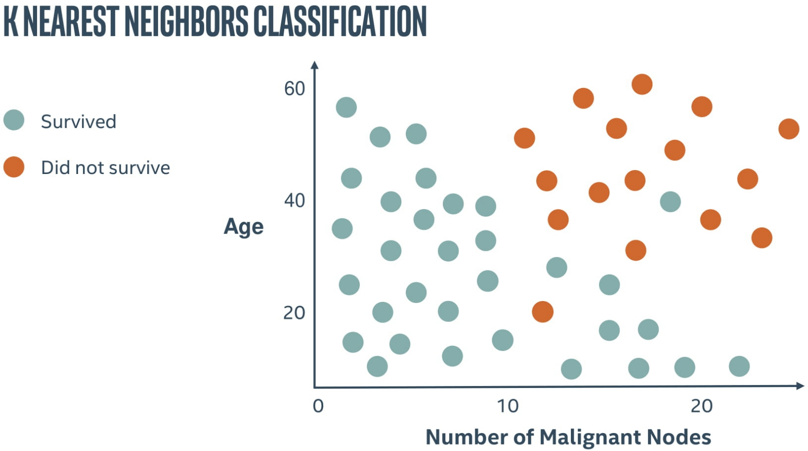
WHAT IS NEEDED FOR CLASSIFICATION?

- **Model data with:**
  - Features that can be quantitated
  - Labels that are known

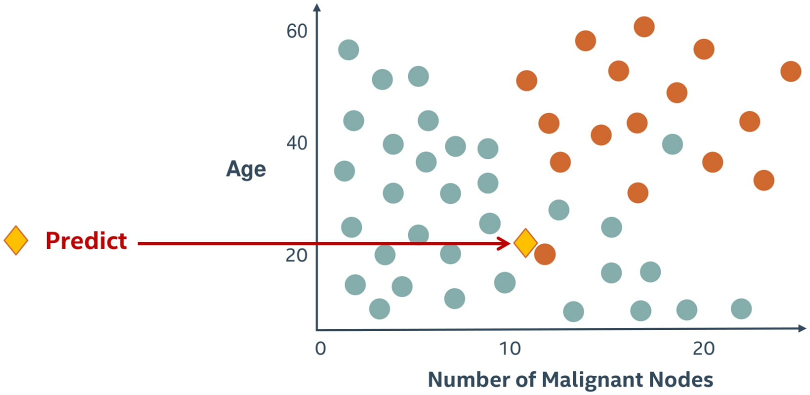
WHAT IS NEEDED FOR CLASSIFICATION?

- **Model data with:**
  - Features that can be quantitated
  - Labels that are known
- **Method to measure similarity**

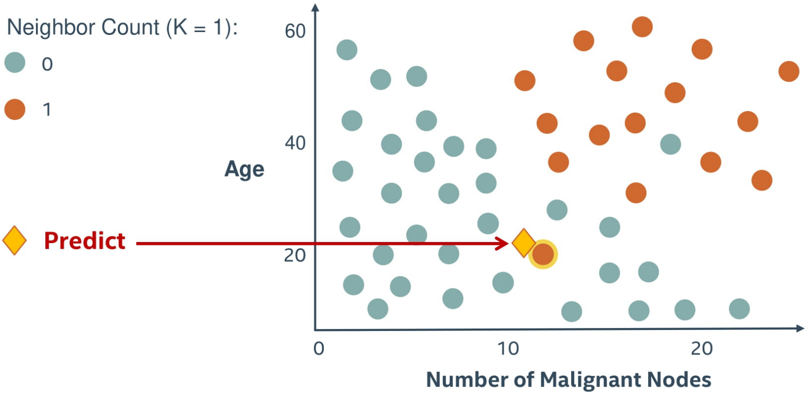
K NEAREST NEIGHBORS CLASSIFICATION


K NEAREST NEIGHBORS CLASSIFICATION
Survived
Did not survive
60
40
20
Age
0
10
20
Number of Malignant Nodes

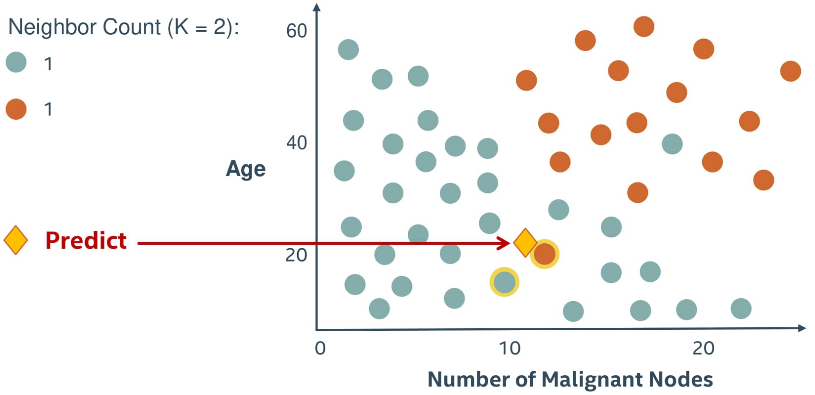
K NEAREST NEIGHBORS CLASSIFICATION



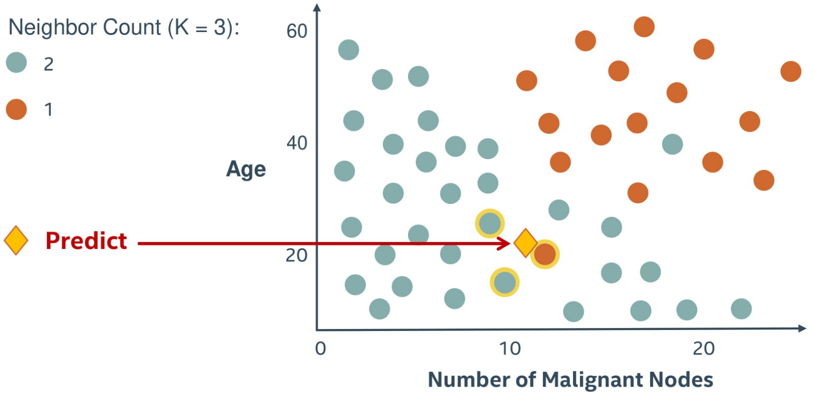
K NEAREST NEIGHBORS CLASSIFICATION



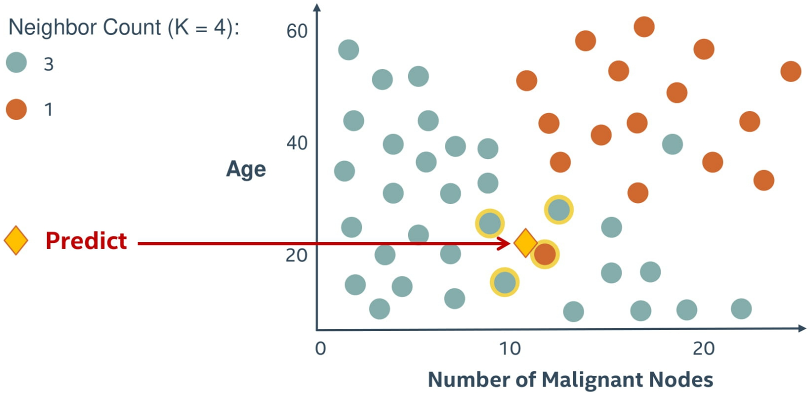
K NEAREST NEIGHBORS CLASSIFICATION



K NEAREST NEIGHBORS CLASSIFICATION



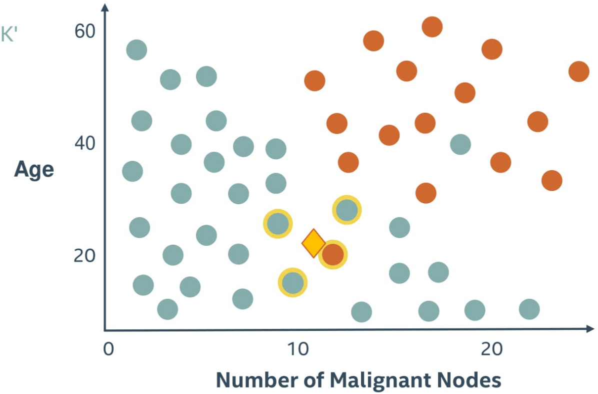
K NEAREST NEIGHBORS CLASSIFICATION



WHAT IS NEEDED TO SELECT A KNN MODEL?

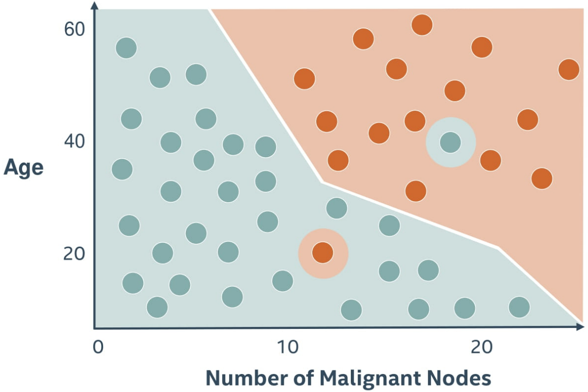
WHAT IS NEEDED TO SELECT A KNN MODEL?

- Correct value for 'K'
- How to measure closeness of neighbors?



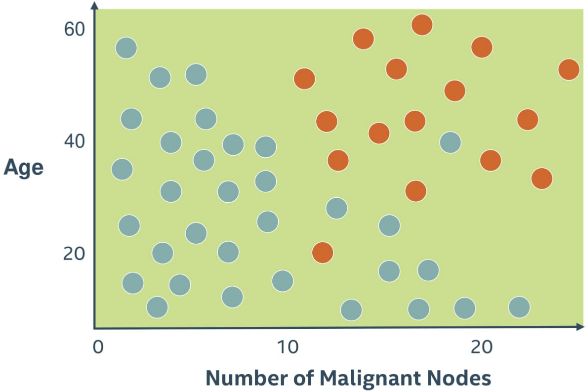
K NEAREST NEIGHBORS DECISION BOUNDARY

K=1

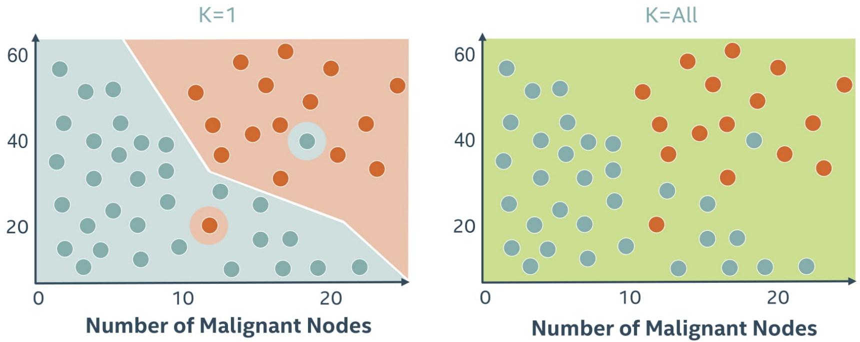


K NEAREST NEIGHBORS DECISION BOUNDARY

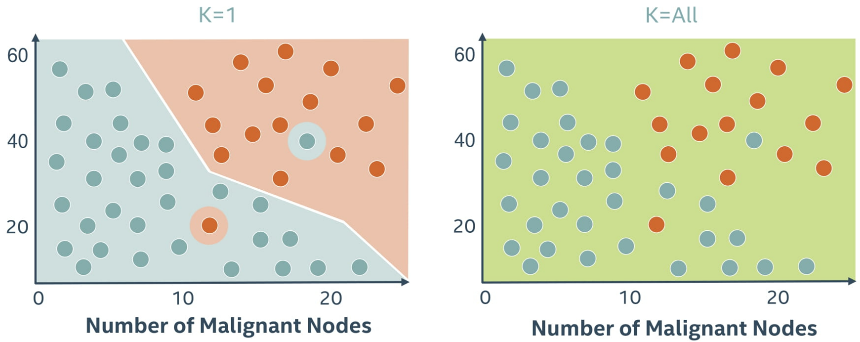
K = All



VALUE OF 'K' AFFECTS DECISION BOUNDARY

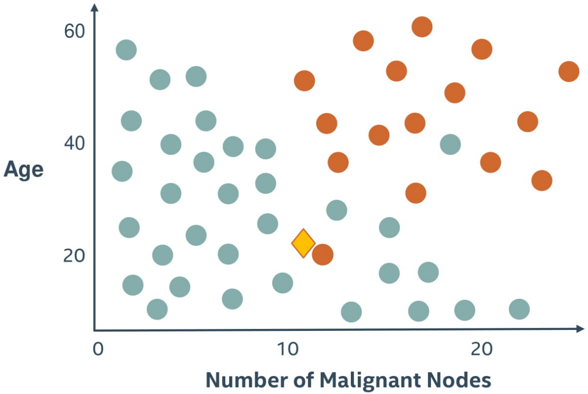


VALUE OF 'K' AFFECTS DECISION BOUNDARY

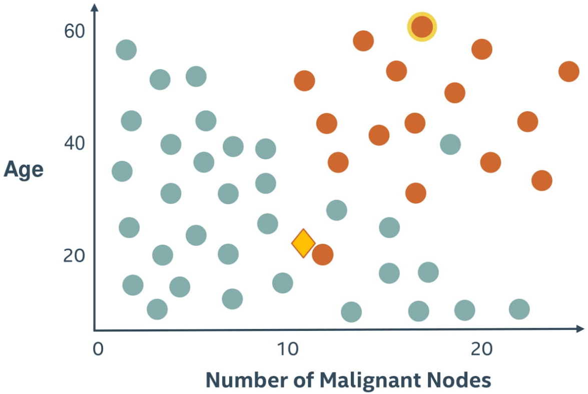


Methods for determining 'K' will be discussed in next lesson

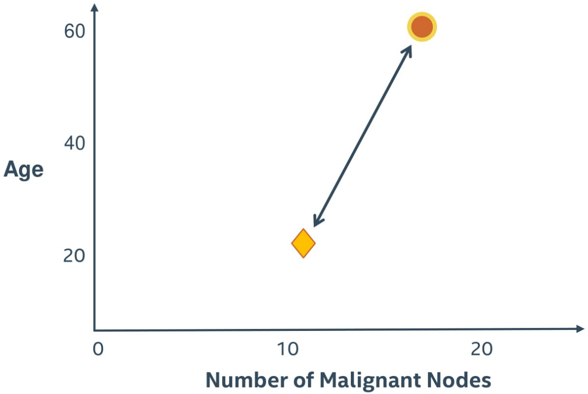
MEASUREMENT OF DISTANCE IN KNN



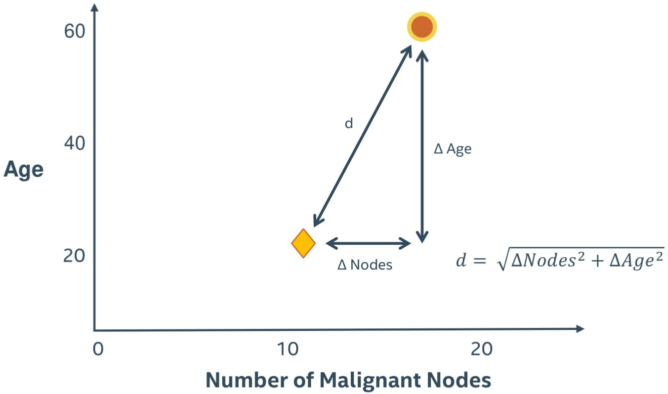
MEASUREMENT OF DISTANCE IN KNN



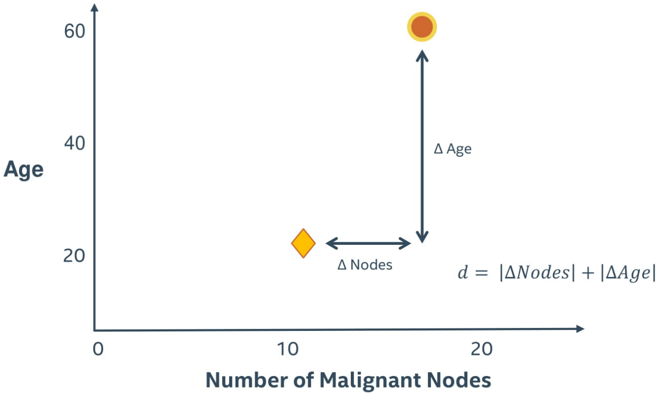
EUCLIDEAN DISTANCE



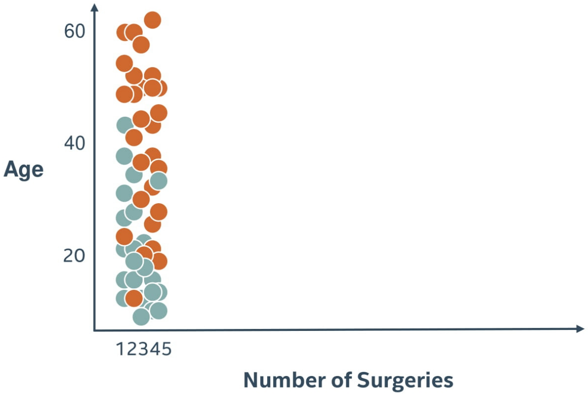
EUCLIDEAN DISTANCE (L2 DISTANCE)



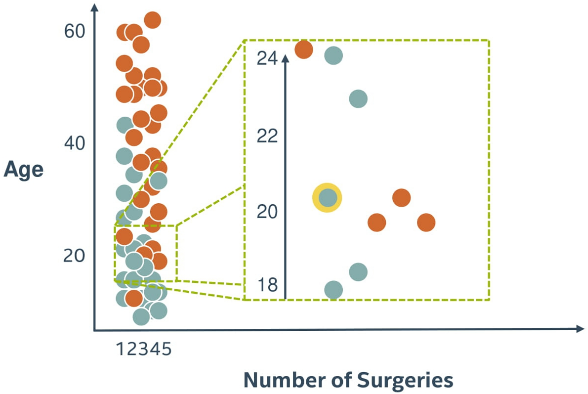
MANHATTAN DISTANCE (L1 OR CITY BLOCK DISTANCE)



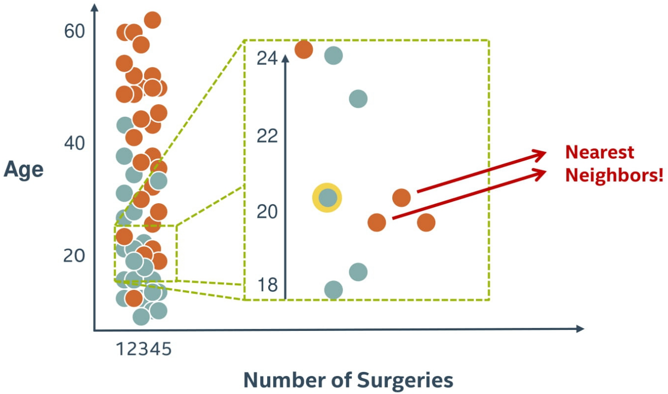
SCALE IS IMPORTANT FOR DISTANCE MEASUREMENT



SCALE IS IMPORTANT FOR DISTANCE MEASUREMENT

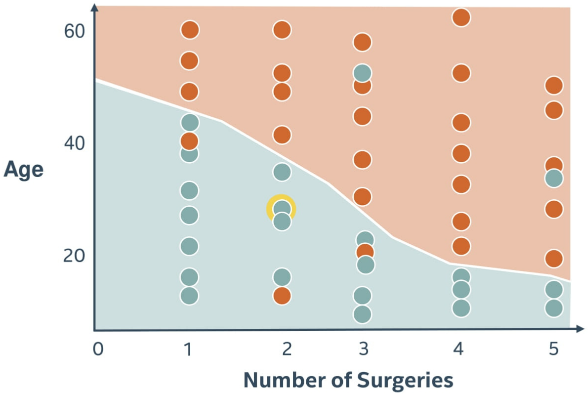


SCALE IS IMPORTANT FOR DISTANCE MEASUREMENT



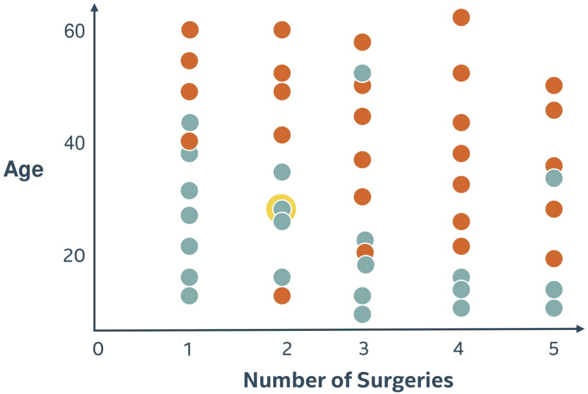
SCALE IS IMPORTANT FOR DISTANCE MEASUREMENT

"Feature Scaling"



SCALE IS IMPORTANT FOR DISTANCE MEASUREMENT

"Feature Scaling"



COMPARISON OF FEATURE SCALING METHODS

- **Standard Scaler:** Mean center data and scale to unit **variance**
- **Minimum-Maximum Scaler:** Scale data to fixed range (usually 0–1)
- **Maximum Absolute Value Scaler:** Scale maximum absolute value

FEATURE SCALING: THE SYNTAX

Import the class containing the scaling method

```
from sklearn.preprocessing import StandardScaler
```

FEATURE SCALING: THE SYNTAX

Import the class containing the scaling method

```
from sklearn.preprocessing import StandardScaler
```

Create an instance of the class

```
StdSc = StandardScaler()
```

FEATURE SCALING: THE SYNTAX

Import the class containing the scaling method

```
from sklearn.preprocessing import StandardScaler
```

Create an instance of the class

```
StdSc = StandardScaler()
```

Fit the scaling parameters and then transform the data

```
StdSc = StdSc.fit(X_data)

X_scaled = KNN.transform(X_data)
```

FEATURE SCALING: THE SYNTAX

Import the class containing the scaling method

```
from sklearn.preprocessing import StandardScaler
```

Create an instance of the class

```
StdSc = StandardScaler()
```

Fit the scaling parameters and then transform the data

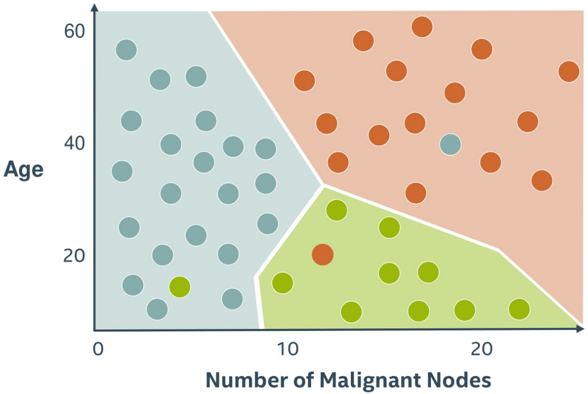
```
StdSc = StdSc.fit(X_data)

X_scaled = KNN.transform(X_data)
```

Other scaling methods exist: MinMaxScaler, MaxAbsScaler.

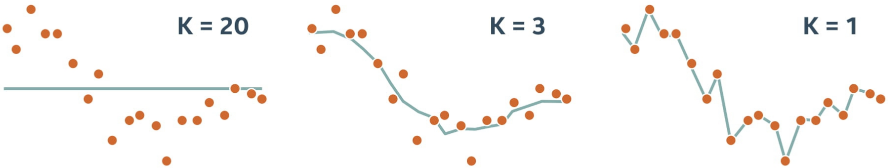
MULTICLASS KNN DECISION BOUNDARY

K=5

- Full remission
- Partial remission
- Did not survive



REGRESSION WITH KNN



CHARACTERISTICS OF A KNN MODEL

- **Fast to create model because it simply stores data**
- **Slow to predict because many distance calculations**
- **Can require lots of memory if data set is large**

K NEAREST NEIGHBORS: THE SYNTAX

Import the class containing the classification method

```
from sklearn.neighbors import KNeighborsClassifier
```

K NEAREST NEIGHBORS: THE SYNTAX

Import the class containing the classification method

```
from sklearn.neighbors import KNeighborsClassifier
```

Create an instance of the class

```
KNN = KNeighborsClassifier(n_neighbors=3)
```

K NEAREST NEIGHBORS: THE SYNTAX

Import the class containing the classification method

```
from sklearn.neighbors import KNeighborsClassifier
```

Create an instance of the class

```
KNN = KNeighborsClassifier(n_neighbors=3)
```

Fit the instance on the data and then predict the expected value

```
KNN = KNN.fit(X_data, y_data)

y_predict = KNN.predict(X_data)
```

K NEAREST NEIGHBORS: THE SYNTAX

Import the class containing the classification method

```
from sklearn.neighbors import KNeighborsClassifier
```

Create an instance of the class

```
KNN = KNeighborsClassifier(n_neighbors=3)
```

Fit the instance on the data and then predict the expected value

```
KNN = KNN.fit(X_data, y_data)

y_predict = KNN.predict(X_data)
```

The **fit** and **predict/transform** syntax will show up throughout the course.

K NEAREST NEIGHBORS: THE SYNTAX

Import the class containing the classification method

```
from sklearn.neighbors import KNeighborsClassifier
```

Create an instance of the class

```
KNN = KNeighborsClassifier(n_neighbors=3)
```

Fit the instance on the data and then predict the expected value

```
KNN = KNN.fit(X_data, y_data)

y_predict = KNN.predict(X_data)
```

Regression can be done with KNeighborsRegressor.

References

Intel Nervana AI Academy
https://software.intel.com/content/www/us/en/develop/training

Questions