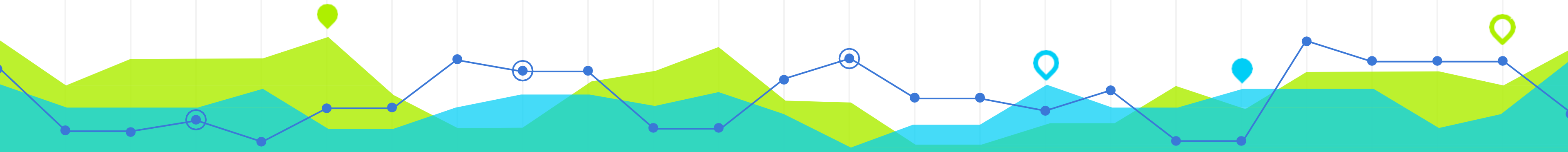
Introduction to Urban Big Data and Machine Learning



Lecture 15 Machine Learning (I)

Wenzheng Li

Announcement

- All assignments (1-4) should be submitted no later than Friday.
- Friday Afternoon: 7-10 minutes presentation
- Final poster due: Sunday night

OUTLINE

- Introduction to Machine Learning
    - What is machine learning?
    - Machine learning types

- Supervised Learning
    - Classification and Regression
    - Understand machine learning
    - Machine learning vs. statistics

1

What is Machine Learning

What is Machine Learning?

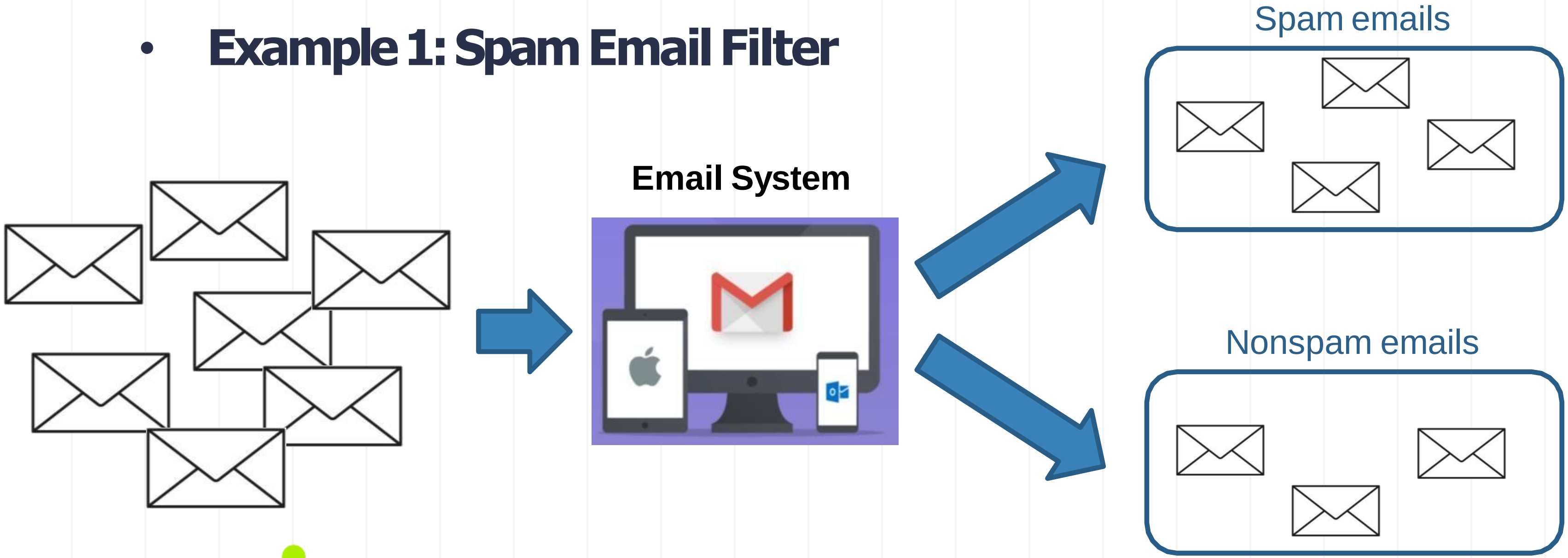
[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

—Arthur Samuel, 1959

Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of Research and Development. ***3*** *(3): 210–229.*

What is Machine Learning?

- **Example 1: Spam Email Filter**



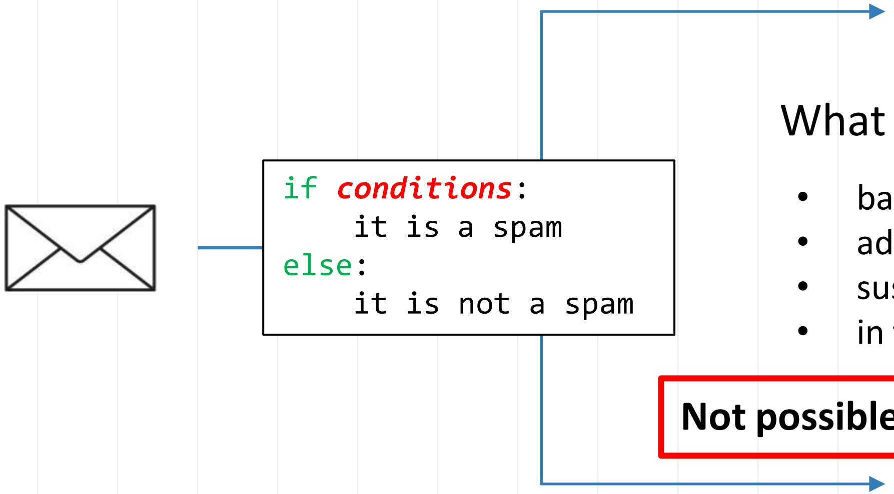
Email System

Spam emails

Nonspam emails

What would you do to create such a system (machine)?

- **Task**: filter spam emails



```
if conditions:
    it is a spam
else:
    it is not a spam
```

Spam emails

Nonspam emails

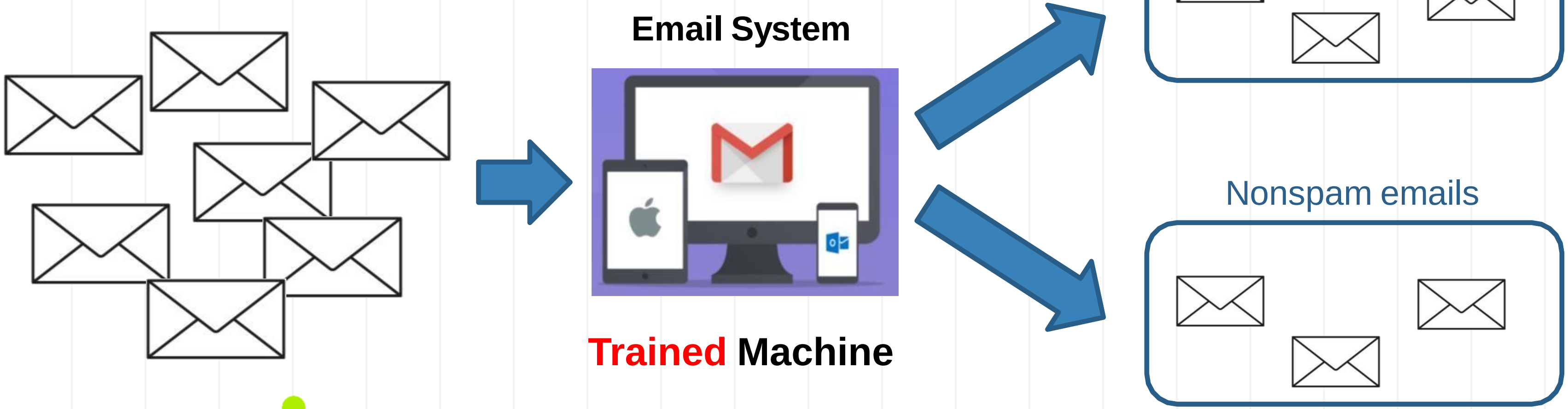
What are the ***conditions***?

- bad words
- ads (what are the ads)?
- suspicious senders?
- in title? In email body?

Not possible to write a rule to exhaust all cases!

What is Machine Learning?

- **Example 1: Spam Email Filter**



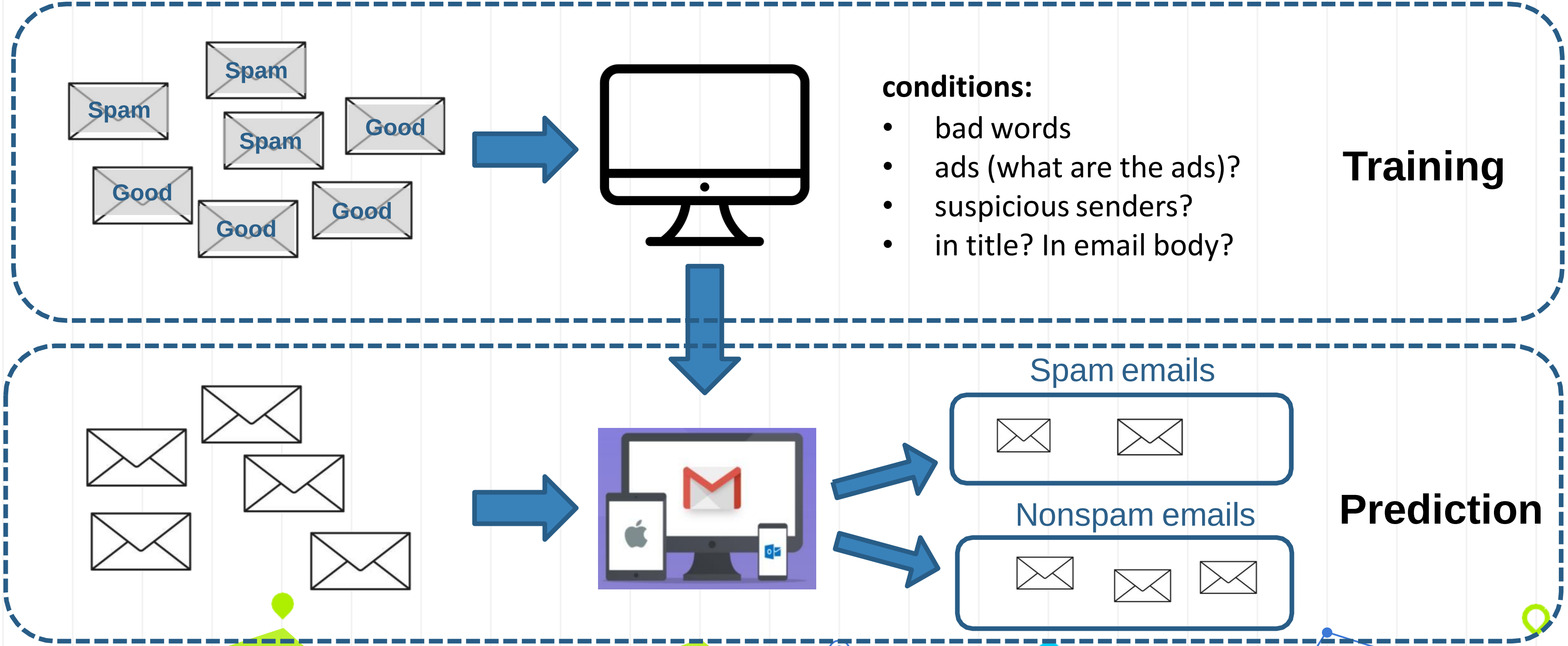
Spam emails

Email System

Trained Machine

Nonspam emails

A spam filter based on machine learning



conditions:

- bad words
- ads (what are the ads)?
- suspicious senders?
- in title? In email body?

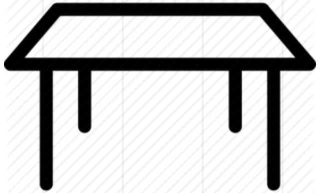
Training

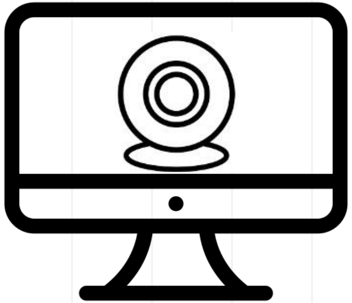
Prediction

What is Machine Learning?

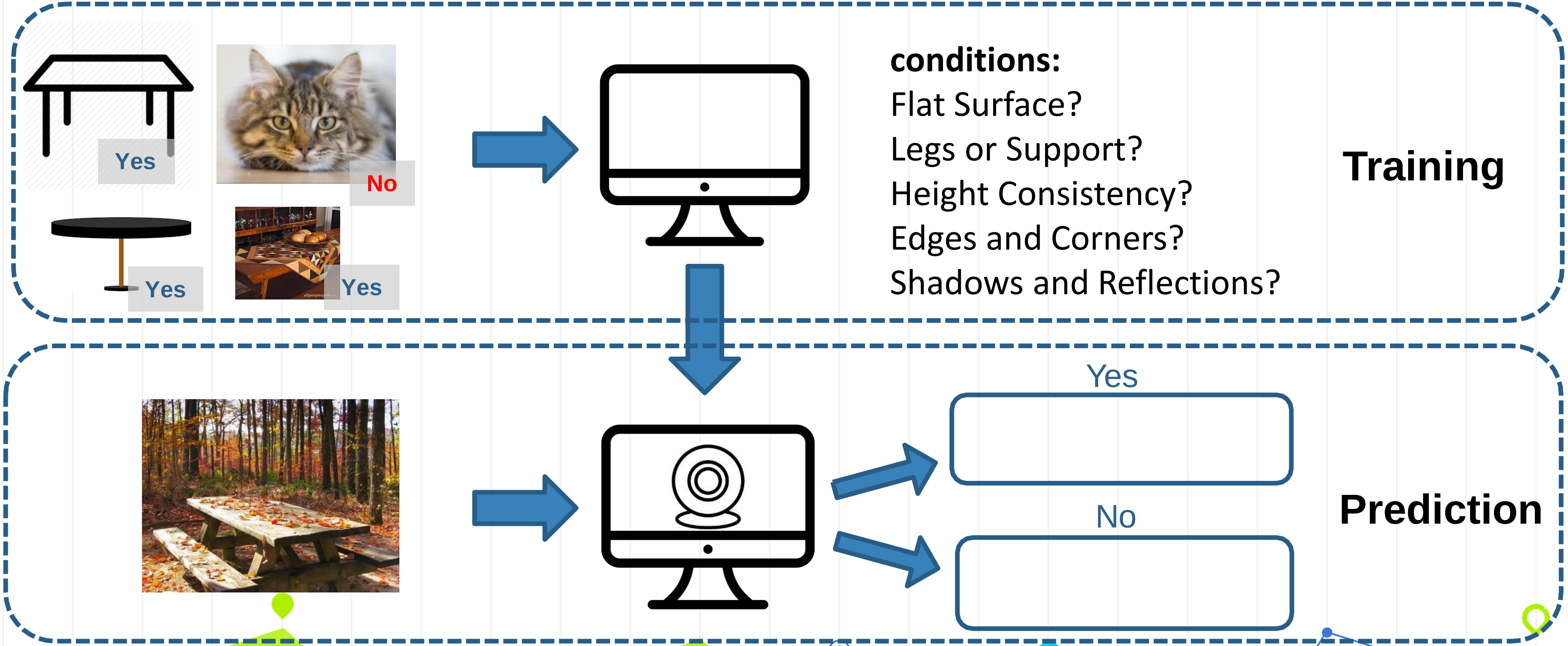
- **Example 2: Image Recognition**

  Task: whether there is a table in the image





A table detector based on machine learning



Yes

No

Yes

Yes

conditions:

Flat Surface?

Legs or Support?

Height Consistency?

Edges and Corners?

Shadows and Reflections?

Training

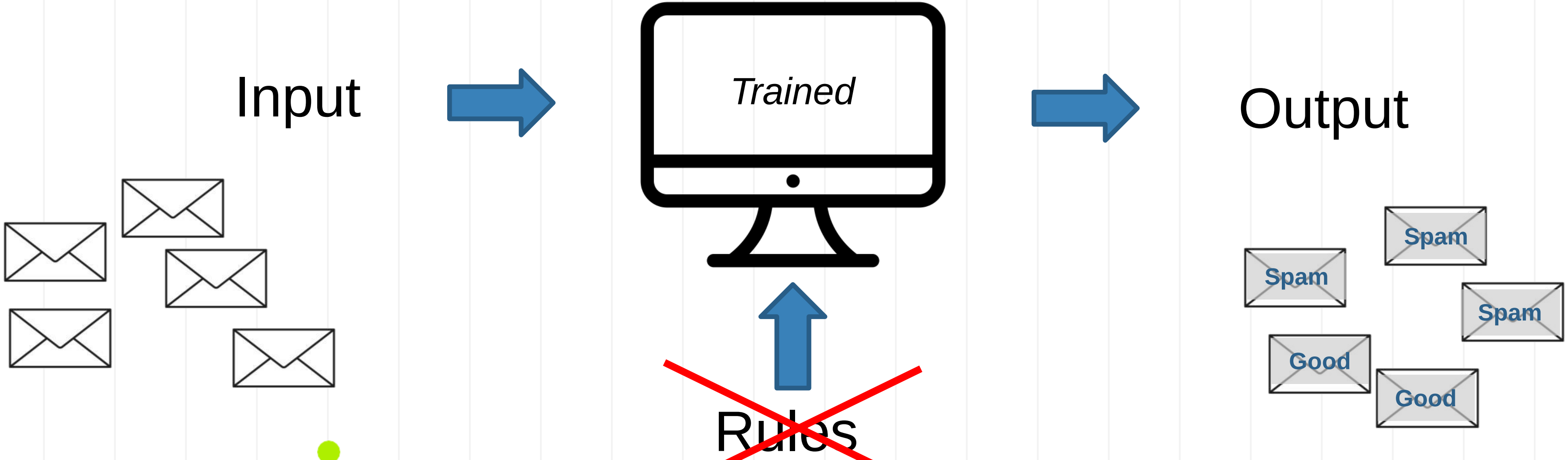
Yes

No

Prediction

What is Machine Learning?

- We want a machine which can predict output accurately (most of time) based on input data.



Input

Trained

Output

~~Rules~~

Spam

Spam

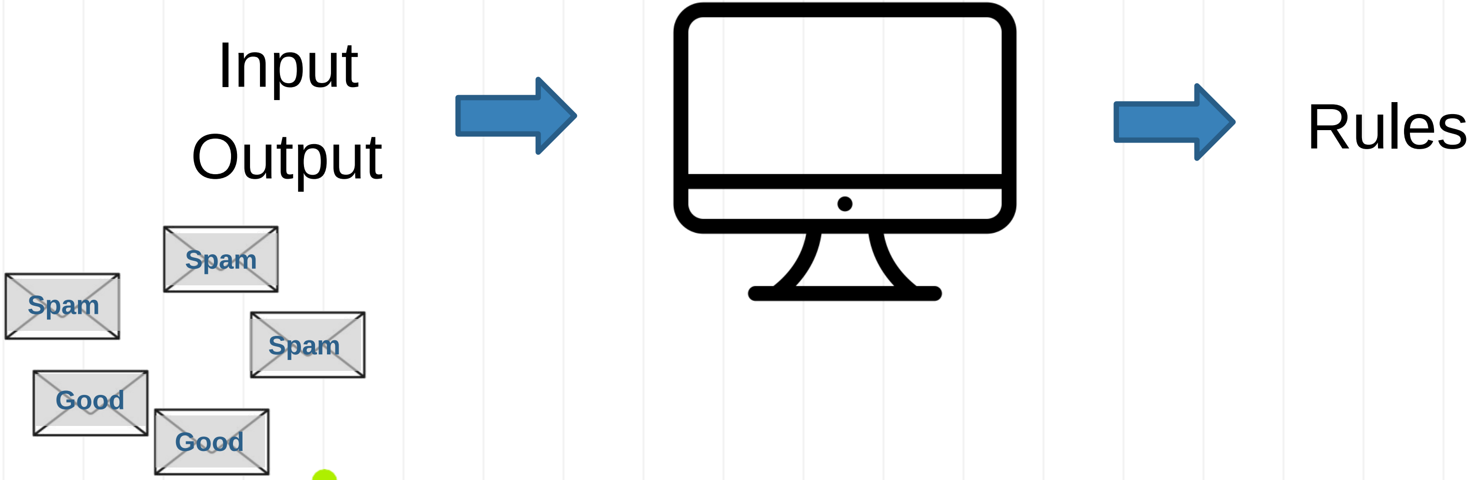
Spam

Good

Good

What is Machine Learning?

- Training



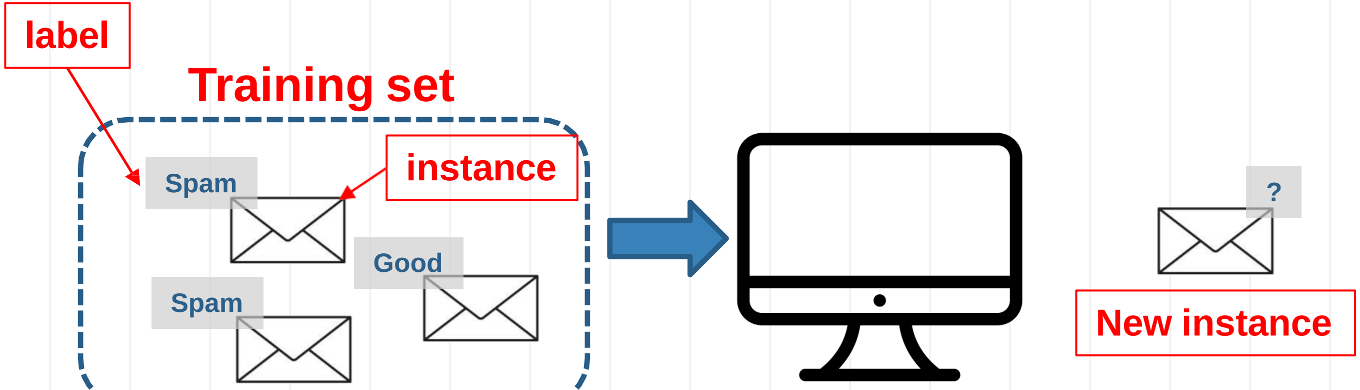
Input

Output

Rules

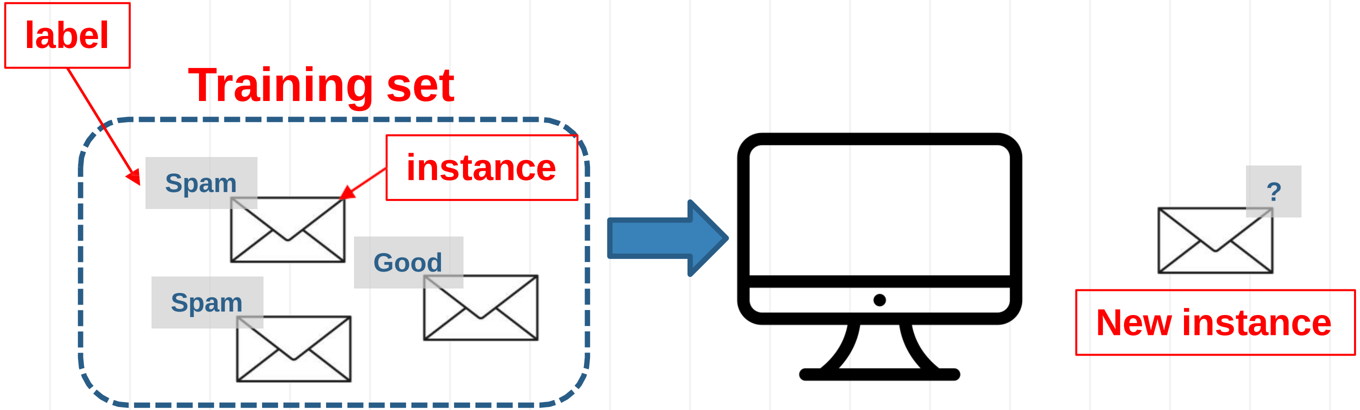
What is Machine Learning?

- Training



Supervised Learning

- Training



1. Supervised Learning: machine is trained with human supervision with a "teacher", (the training set is labeled)

Then, what is unsupervised learning?

1. Supervised Learning: machine is trained with human supervision with a "teacher", (the training set is labeled)

2. Unsupervised Learning: machine is trained without human supervision without a "teacher", (the training set is not labeled)

Unsupervised Learning

- We want to classify customers into different types based on their attributes

Unsupervised Learning

- We want to classify customers into different types based on their attributes
- Clustering data based on identified patterns or structures.

Input

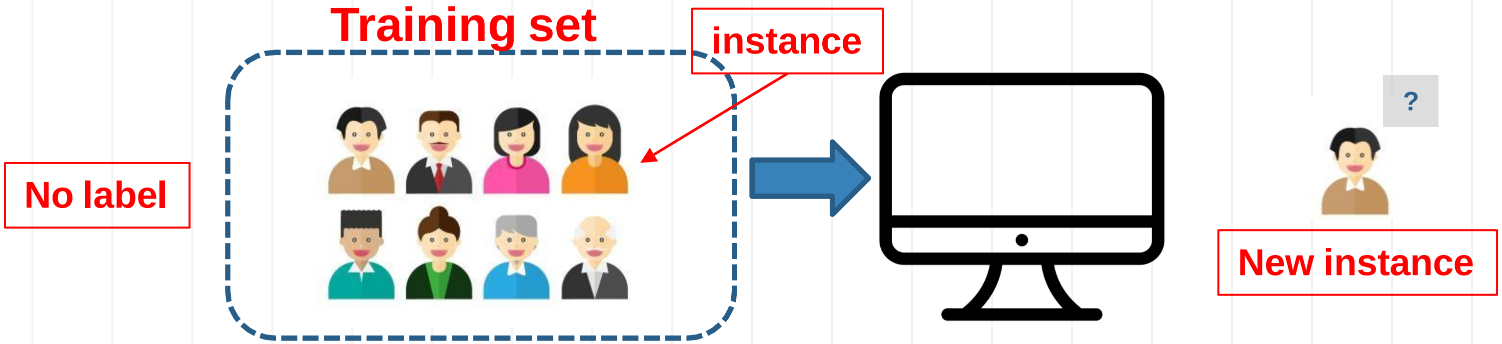
Trained

Output

We do not know the output!

Unsupervised Learning

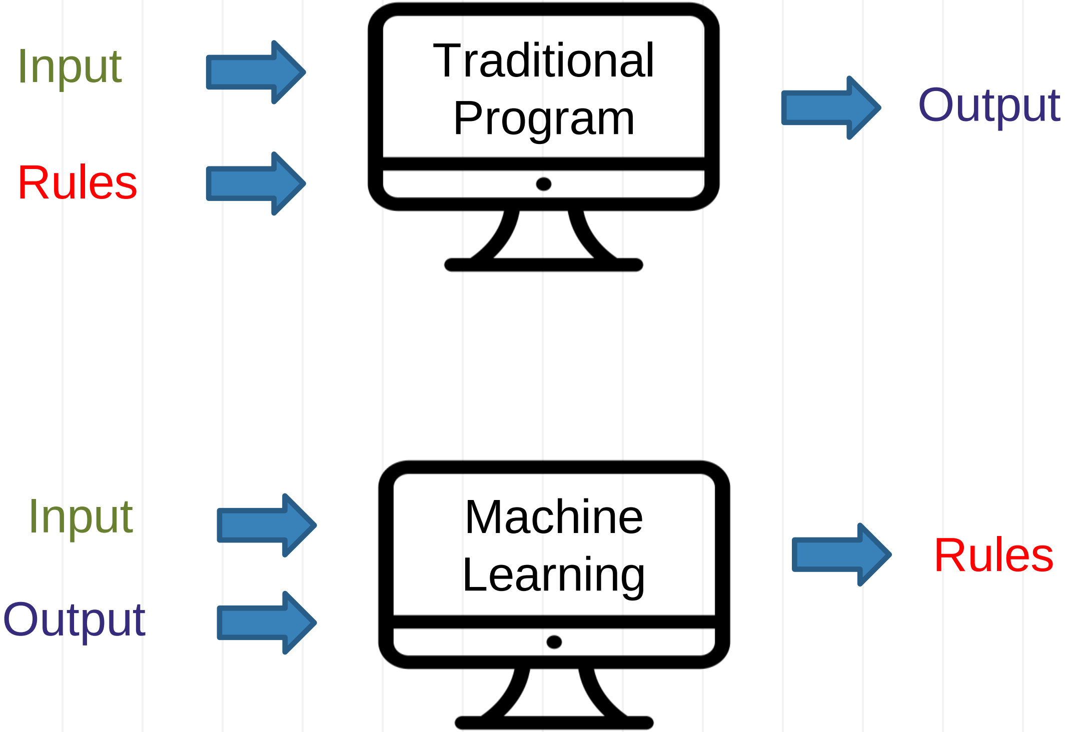
- Training



2. Unsupervised Learning: machine is trained without human supervision without a “teacher”, (the training set is not labeled)

Machine Learning Types

1. Supervised Learning: machine is trained with human supervision with a “teacher”, (the training set is labeled)
2. Unsupervised Learning: machine is trained without human supervision without a “teacher”, (the training set is not labeled)
3. Semisupervised Learning
4. Reinforcement Learning



Input

Rules

Traditional Program

Output

Input

Output

Machine Learning

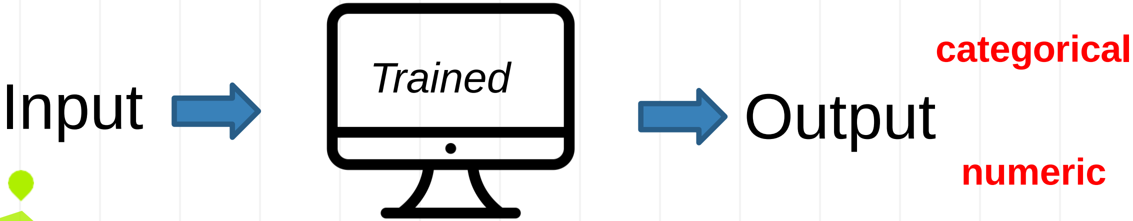
Rules

Supervised Learning

2

Supervised Learning

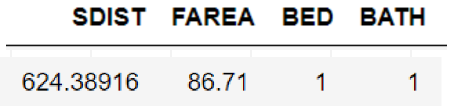
- Associating features with some label.
- **Two tasks :**
  - Classification: labels as discrete categories
    - Spam email or not
    - With a table or not
  - Regression: labels as continuous quantities.
    - Housing price



Understand Machine Learning

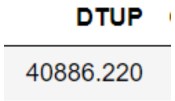
- What do I mean if I want to build a machine learning model for my housing transaction dataset?

Housing datasets with attributes



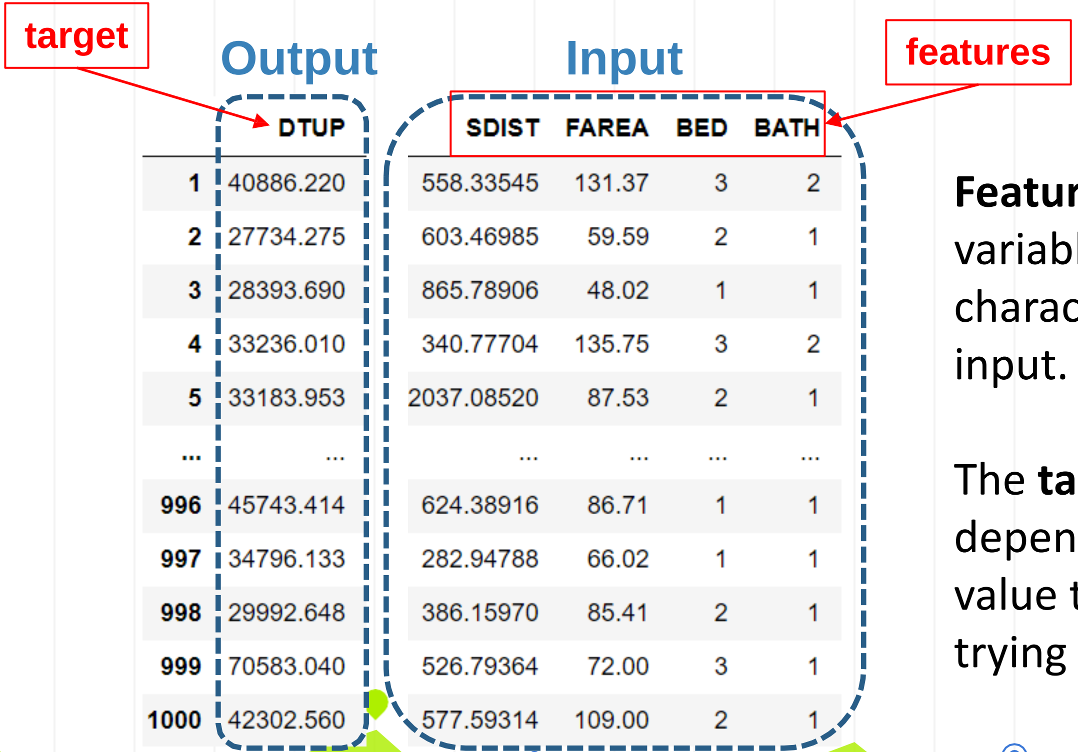
SDIST	FAREA	BED	BATH
624.38916	86.71	1	1

Housing prices



DTUP
40886.220

Understand Machine Learning

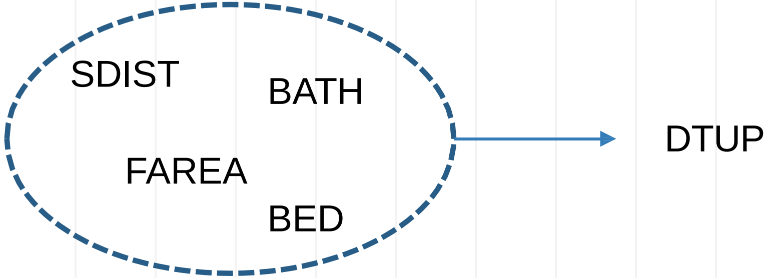


	DTUP	SDIST	FAREA	BED	BATH
1	40886.220	558.33545	131.37	3	2
2	27734.275	603.46985	59.59	2	1
3	28393.690	865.78906	48.02	1	1
4	33236.010	340.77704	135.75	3	2
5	33183.953	2037.08520	87.53	2	1
...	...	...	...	...	...
996	45743.414	624.38916	86.71	1	1
997	34796.133	282.94788	66.02	1	1
998	29992.648	386.15970	85.41	2	1
999	70583.040	526.79364	72.00	3	1
1000	42302.560	577.59314	109.00	2	1

Features, also known as attributes or variables, are the individual measurable characteristics of the data that are used as input.

The **target**, also known as the label or dependent variable, is the outcome or value that the machine learning model is trying to predict.

Understand Machine Learning



SDIST

BATH

FAREA

BED

DTUP

- How about a linear regression model?
  - Linear regression is a type of machine learning model

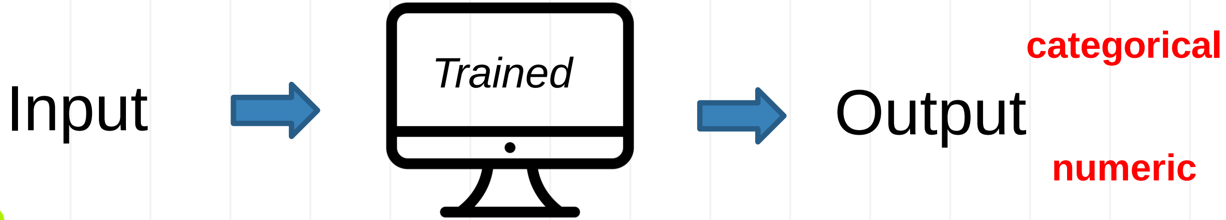
$$\boldsymbol{DTUP} = \beta_0 + \beta_1\boldsymbol{SDIST} + \beta_2\boldsymbol{FAREA} + \beta_3\boldsymbol{BED} + \beta_4\boldsymbol{BATH} + \varepsilon$$

Supervised Learning

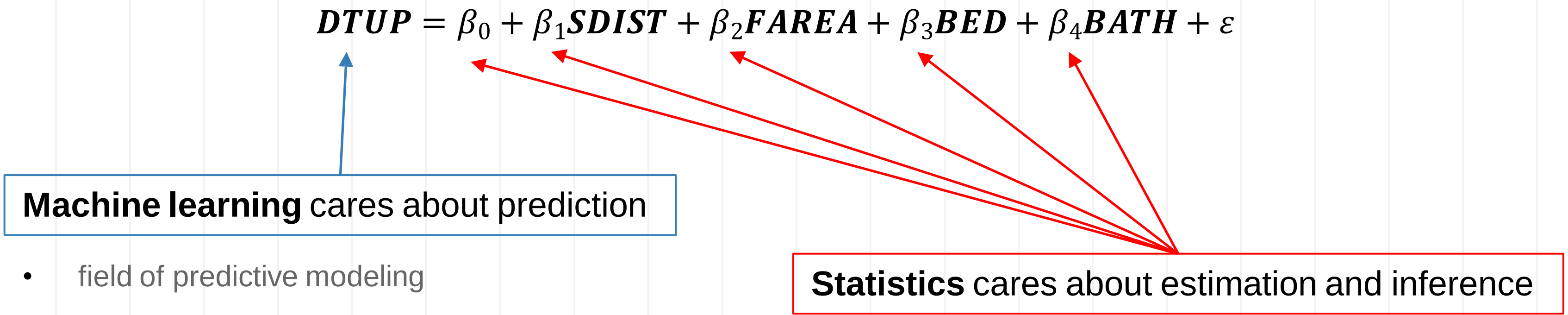
- **Two tasks :**
  - Classification: output is **categorical**
    - Spam email or not
    - With a table or not

    Logistic Regression is here!!!
  - Regression: output is **numeric**
    - Housing price

    Linear Regression is here!!!



Machine Learning vs. Statistics



$$\boldsymbol{DTUP} = \beta_0 + \beta_1\boldsymbol{SDIST} + \beta_2\boldsymbol{FAREA} + \beta_3\boldsymbol{BED} + \beta_4\boldsymbol{BATH} + \varepsilon$$

Machine learning cares about prediction

- field of predictive modeling
- concerned with minimizing the prediction error or making the most accurate predictions
- borrow algorithms from statistics for predication purposes

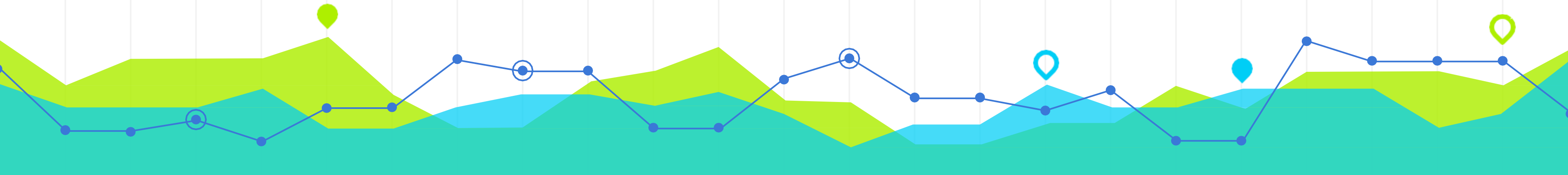
Statistics cares about estimation and inference

- field of statistical modeling
- understanding the relationship between variables
- many models/algorithms have been borrowed by machine learning.

Machine Learning vs. Statistics

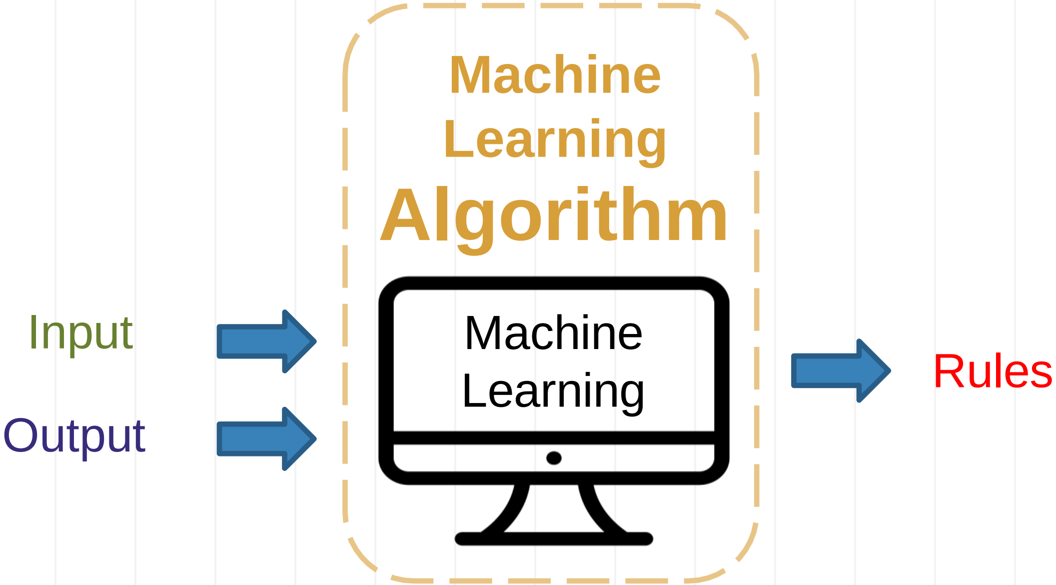
- **Statistics**: hypothesis testing, model assumptions, explanation, and interpretation

- In **machine learning**, different than in applied statistics, we are less interested in what these parameters are, and more in how well they can
  - Make **predictions**
  - Describe **underlying structures or characteristics in the data**



3

Machine Learning Algorithm



Input

Output

Machine Learning Algorithm

Machine Learning

Rules

Algorithm

- In mathematics and computer science, an **algorithm** is a finite sequence of well–defined, computer–implementable **instructions**, typically to solve a class of problems or to perform a computation. (Wikipedia)

Machine Learning Algorithm

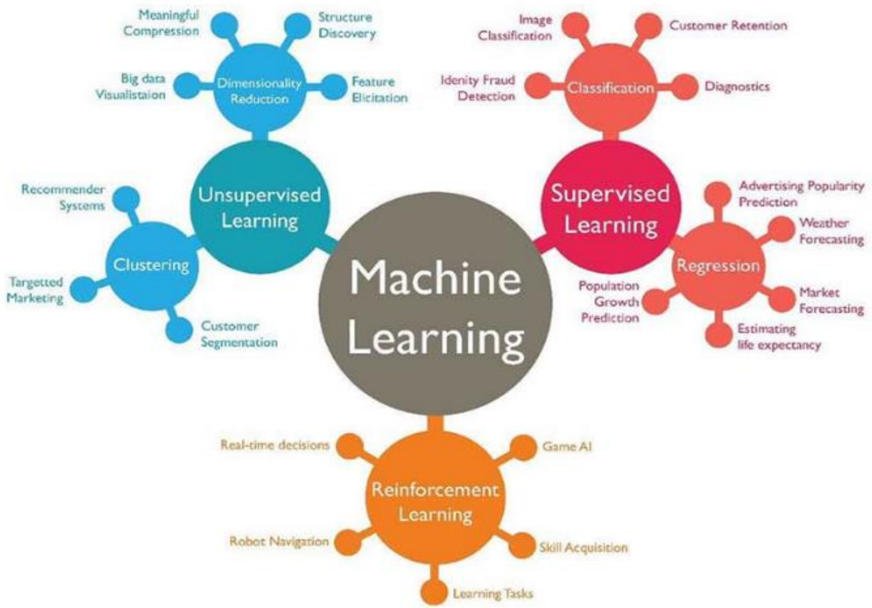
- **Supervised learning**
  - Classification
    - K-nearest neighbors
    - logistic regression
    - Random forest

    …
  - Regression
    - Linear regression

    …
- **Unsupervised learning**
  - Clustering
    - K-means clustering

    …
  - Dimensionality Reduction
    - PCA and factor analysis

    …



Algorithm Example: Linear Regression

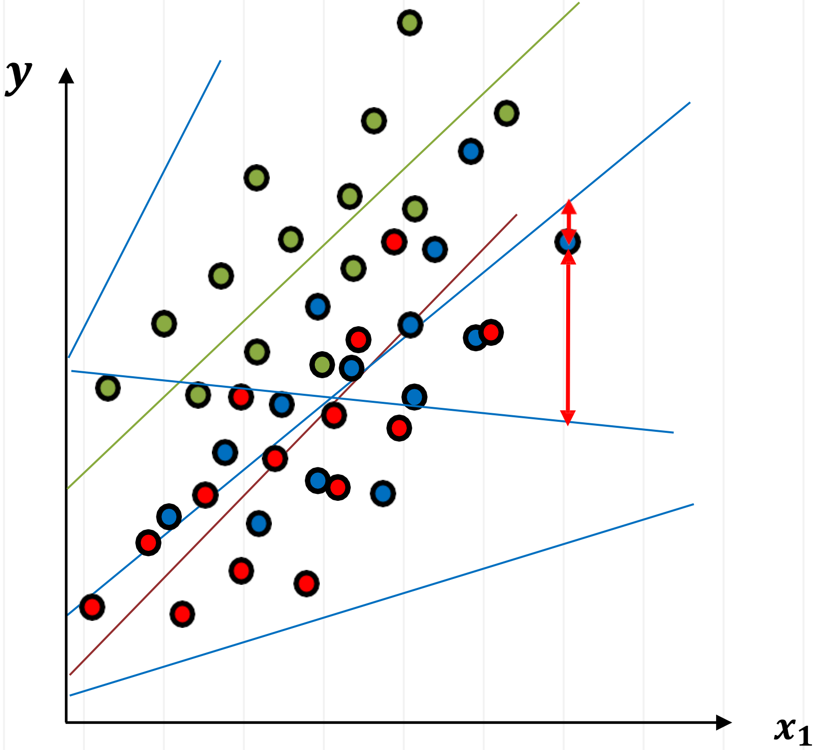
- **Supervised learning**
    - Regression Problem

y	x_1
20000	800
60000	2000
14000	632
54000	1800
20000	750
20000	400
…	…

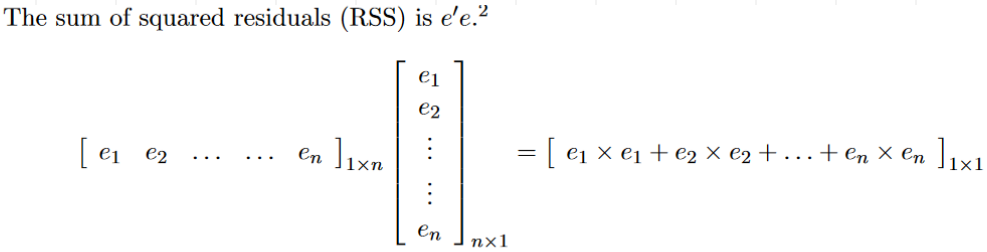
$$y_i = \beta_0 + \beta_1 x_{1i} + \varepsilon_i$$

OLS: Ordinary Least Squares

minimize sum of squared errors

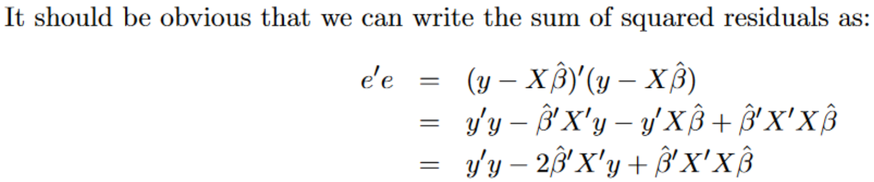


Algorithm Example: Linear Regression



The sum of squared residuals (RSS) is $e'e$.[2]

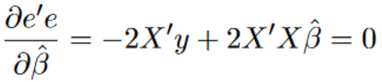
$$\left[\begin{array}{ccccc} e_1 & e_2 & \ldots & \ldots & e_n \end{array}\right]_{1\times n} \left[\begin{array}{c} e_1 \\ e_2 \\ \vdots \\ \vdots \\ e_n \end{array}\right]_{n\times 1} = \left[\begin{array}{c} e_1\times e_1 + e_2\times e_2 + \ldots + e_n\times e_n \end{array}\right]_{1\times 1}$$



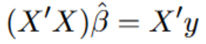
It should be obvious that we can write the sum of squared residuals as:

$$\begin{aligned} e'e &= (y - X\hat{\beta})'(y - X\hat{\beta}) \\ &= y'y - \hat{\beta}'X'y - y'X\hat{\beta} + \hat{\beta}'X'X\hat{\beta} \\ &= y'y - 2\hat{\beta}'X'y + \hat{\beta}'X'X\hat{\beta} \end{aligned}$$

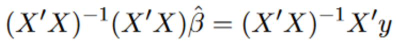
We need to take the derivative of the above equation:



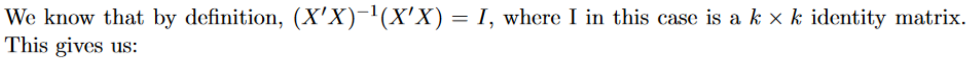
$$\frac{\partial e'e}{\partial\hat{\beta}} = -2X'y + 2X'X\hat{\beta} = 0$$



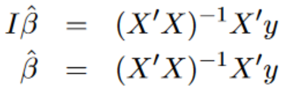
$$(X'X)\hat{\beta} = X'y$$



$$(X'X)^{-1}(X'X)\hat{\beta} = (X'X)^{-1}X'y$$



We know that by definition, $(X'X)^{-1}(X'X) = I$, where I in this case is a $k\times k$ identity matrix. This gives us:



$$\begin{aligned} I\hat{\beta} &= (X'X)^{-1}X'y \\ \hat{\beta} &= (X'X)^{-1}X'y \end{aligned}$$

https://web.stanford.edu/~mrosenfe/soc_meth_proj3/matrix_OLS_NYU_notes.pdf

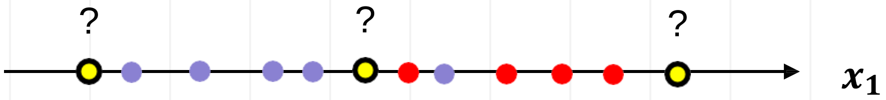
Algorithm Example: Classification

● y = 0

● y = 1

- **Supervised learning**
  - Classification Problem
    - Binary Choice

y	x_1
0	800
1	2000
0	632
1	1800
1	750
0	400
…	…



Algorithm Example: Logistic Regression

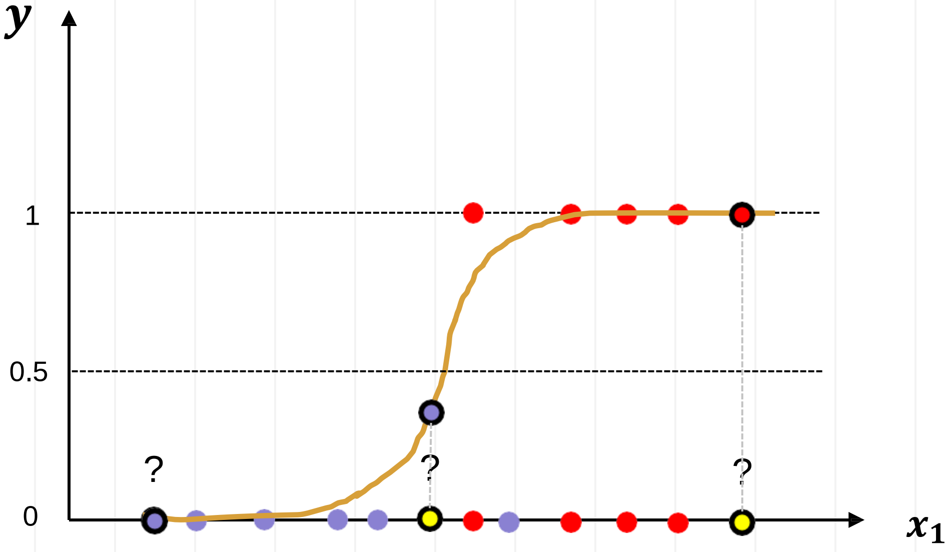
● y = 0
● y = 1

- **Supervised learning**
    - Classification Problem
        - Binary Choice

y	x_1
0	800
1	2000
0	632
1	1800
1	750
0	400
…	…

$$P(y_i = 1|x_{1i}) = \frac{\exp(\beta_0 + \beta_1 x_{1i})}{1 + \exp(\beta_0 + \beta_1 x_{1i})}$$

MLE: Maximum Likelihood Estimation

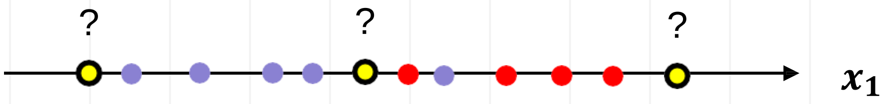


Algorithm Example: K-Nearest Neighbors

● y = 0
● y = 1

- **Supervised learning**
  - ○ Classification Problem
    - ❖ Binary Choice

y	x_1
0	800
1	2000
0	632
1	1800
1	750
0	400
…	…



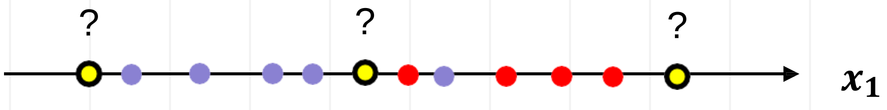
Algorithm Example: K-Nearest Neighbors

● y = 0

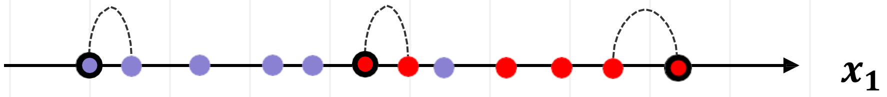
● y = 1

- **Supervised learning**
  - Classification Problem
    - Binary Choice

y	x_1
0	800
1	2000
0	632
1	1800
1	750
0	400
…	…



1 Nearest Neighbor

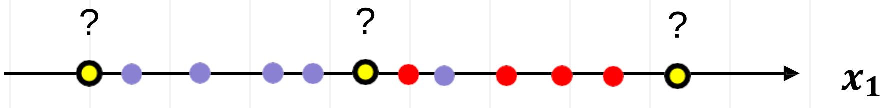


Algorithm Example: K-Nearest Neighbors

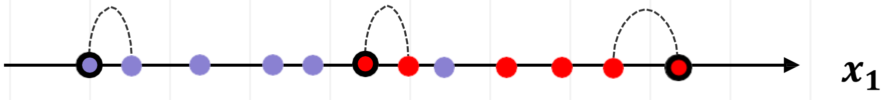
● y = 0
● y = 1

- **Supervised learning**
  - ○ Classification Problem
    - ❖ Binary Choice

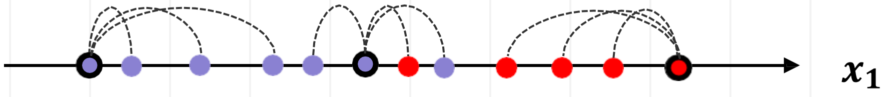
y	x_1
0	800
1	2000
0	632
1	1800
1	750
0	400
…	…



1 Nearest Neighbor



3 Nearest Neighbor



k Nearest Neighbor…

Algorithm Example: K-Nearest Neighbors

- **Supervised learning**
  - Classification Problem
    - Binary Choice

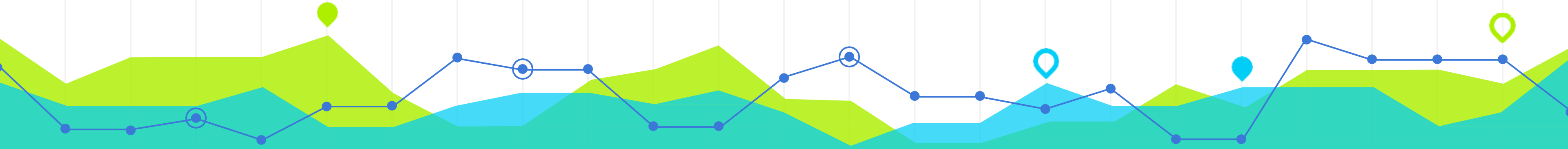
y	x_1
0	800
1	2000
0	632
1	1800
1	750
0	400
…	…

KNN is often referred to as a "lazy learner“

- it simply stores the training data and makes predictions by calculating distances to the known data points

The number of nearest neighbors (k) to consider when making a prediction

- **Small k:** Can lead to a model that is sensitive to noise in the training data (overfitting).
- **Large k:** Can lead to a model that is too generalized (underfitting).



4

Machine Learning Steps

Machine Learning Steps

- Gathering and loading data (what features to collect)
- Exploring data (e.g., pandas and visualization)
- Transforming data (e.g., string to numeric)
- Splitting data for training and testing
- Choosing and creating a model
- Training
- Testing (evaluating accuracy)
- Tuning the model (hyperparameters)
- Making predictions on new data

Data Preparation

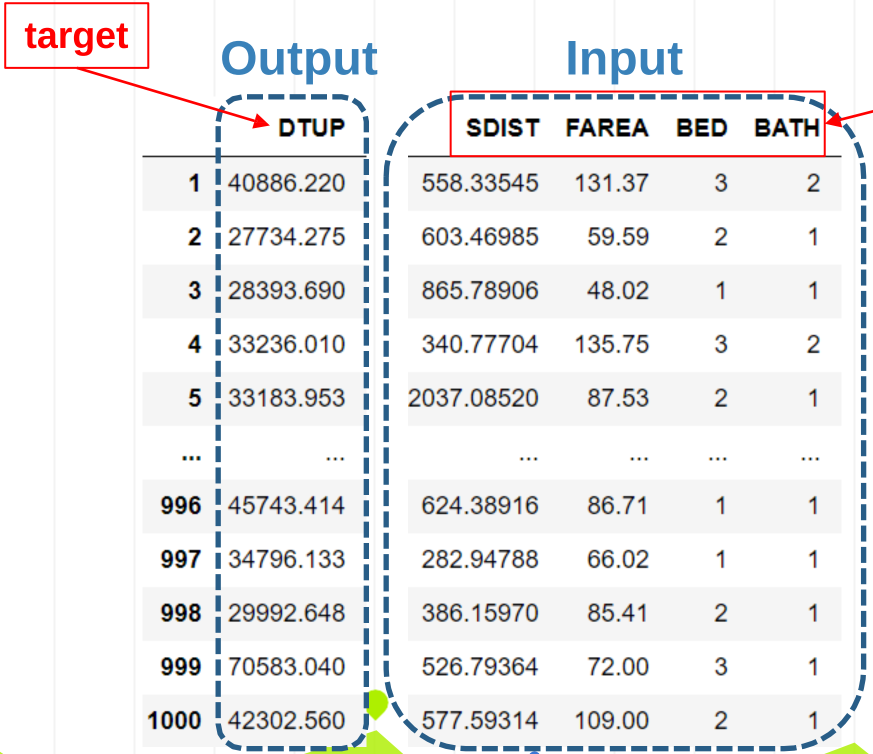
Transforming data (e.g., string to numeric)

Label Encoding: Converts categorical labels into numerical values.
["red", "green", "blue"] -> [0, 1, 2]

One-Hot Encoding: Converts categorical labels into numerical values.
["red", "green", "blue"] -> three dummy variables

Ordinal Encoding: Similar to label encoding but used for ordinal categories where there is an inherent order.
["low", "medium", "high"] -> [1, 2, 3]

Training



	DTUP	SDIST	FAREA	BED	BATH
1	40886.220	558.33545	131.37	3	2
2	27734.275	603.46985	59.59	2	1
3	28393.690	865.78906	48.02	1	1
4	33236.010	340.77704	135.75	3	2
5	33183.953	2037.08520	87.53	2	1
...	...	...	...	...	...
996	45743.414	624.38916	86.71	1	1
997	34796.133	282.94788	66.02	1	1
998	29992.648	386.15970	85.41	2	1
999	70583.040	526.79364	72.00	3	1
1000	42302.560	577.59314	109.00	2	1

Features, also known as attributes or variables, are the individual measurable characteristics of the data that are used as input.

The **target**, also known as the label or dependent variable, is the outcome or value that the machine learning model is trying to predict.

Splitting data for training and testing

Training set

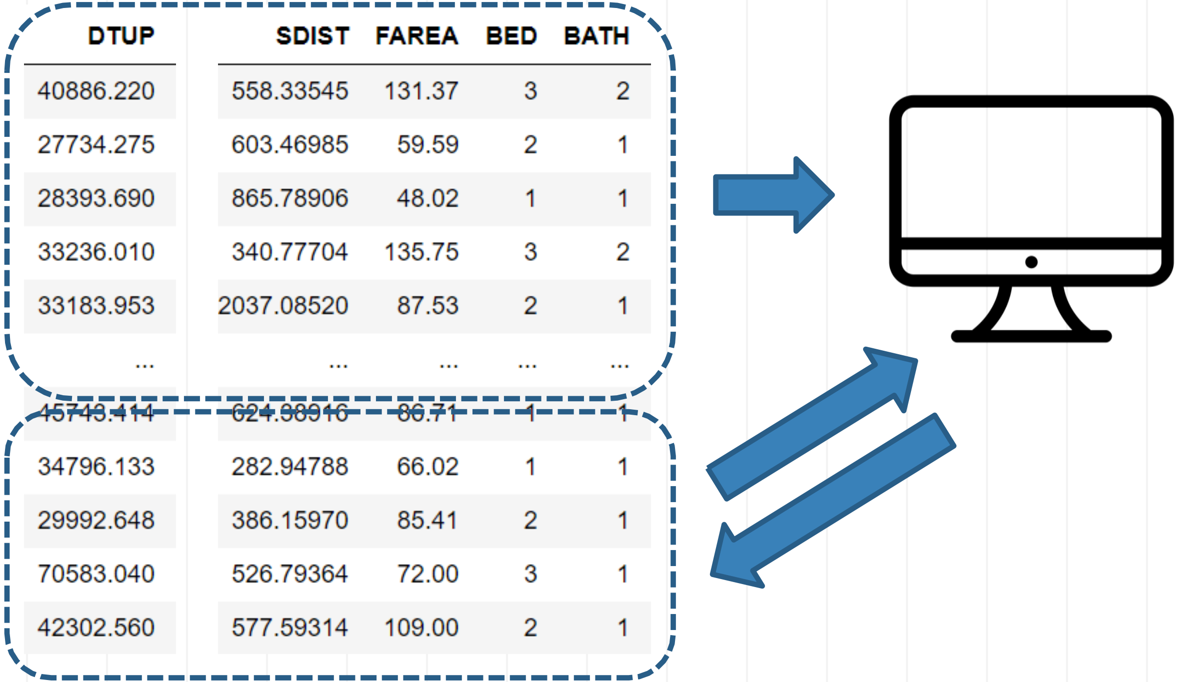
75%

(80%)

Testing set

25%

(20%)



DTUP	SDIST	FAREA	BED	BATH
40886.220	558.33545	131.37	3	2
27734.275	603.46985	59.59	2	1
28393.690	865.78906	48.02	1	1
33236.010	340.77704	135.75	3	2
33183.953	2037.08520	87.53	2	1
...	...	...	...	...
45743.414	624.38916	86.71	1	1
34796.133	282.94788	66.02	1	1
29992.648	386.15970	85.41	2	1
70583.040	526.79364	72.00	3	1
42302.560	577.59314	109.00	2	1

Machine Learning Steps

- Gathering and loading data
- Exploring data (e.g., pandas and visualization)
- Transforming data (e.g., string to numeric)
- Splitting data for training and testing
- Choosing and creating a model
- Training
- Testing (evaluating accuracy) How to measure accuracy?
- Tuning the model (hyperparameters)
- Making predictions on new data

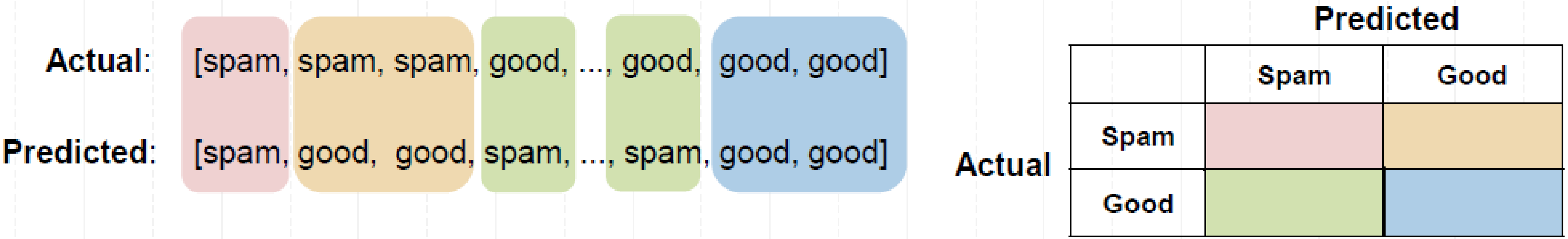
Classification (Supervised Learning)

- **Example**: Spam Email Filter

- We prepared a dataset with 5000 emails (with features and labels)
- We split the dataset into training set (4000, 80%) and testing set (1000, 20%)
- We created a model (e.g., k-nearest neighbors or logistic regression)
- We trained the model using the training set (4000 instances)
- Now, we want to test the model and evaluate the accuracy…

Classification (Supervised Learning)

- **Example:** Spam Email Filter
- Now, we want to test the model and evaluate the accuracy…
  - We predict the labels of the 1000 instances in the testing set
  - Then compare with their actual labels



Actual: [spam, spam, spam, good, ..., good, good, good]

Predicted: [spam, good, good, spam, ..., spam, good, good]

		Predicted	
		Spam	Good
Actual	Spam	🟥	🟧
	Good	🟩	🟦

Confusion Matrix

- Describe the performance of a **classification** model

Predicted

	Total = 1000	**Spam**	**Non-Spam**
Actual	**Spam**	330	70
	Non-Spam	90	510

Predicted

		Spam	**Non-Spam**
Actual	**Spam**	True Positive	False Negative
	Non-Spam	False Positive	True Negative

Type II error

Type I error

Overall, how often is the model correct?

Metric 1: Accuracy

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total}$$

$$= \frac{330 + 510}{1000}$$

$$= 0.84$$

Confusion Matrix

		Predicted	
	Total = 1000	**Spam**	**Non-Spam**
Actual	**Spam**	330	70
	Non-Spam	90	510

		Predicted	
		Spam	**Non-Spam**
Actual	**Spam**	True Positive	False Negative
	Non-Spam	False Positive	True Negative

How often does the model correctly identify positives (spam emails)?

Metric 2: Recall

(Sensitivity or True Positive Rate)

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{330}{330 + 70}$$

$$= 0.825$$

Confusion Matrix

Predicted

	Total = 1000	**Spam**	**Non-Spam**
Actual	**Spam**	330	70
	Non-Spam	90	510

Predicted

		Spam	**Non-Spam**
Actual	**Spam**	True Positive	False Negative
	Non-Spam	False Positive	True Negative

Metric 3: Precision

When the model predicts positive, how often is it correct?

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{330}{330 + 90}$$

$$\approx 0.786$$

Confusion Matrix

How often does the model correctly identify negatives(non-spam emails)?

Predicted

Actual	Total = 1000	Spam	Non-Spam
	Spam	330	70
	Non-Spam	90	510

Predicted

Actual		Spam	Non-Spam
	Spam	True Positive	False Negative
	Non-Spam	False Positive	True Negative

Metric 4: True Negative Rate (Specificity)

$$= \frac{True\ Negative}{True\ Negative + False\ Positive}$$

$$= \frac{510}{510 + 90}$$

$$= 0.85$$

Confusion Matrix

Predicted

	Total = 1000	Spam	Non-Spam
Actual	**Spam**	330	70
	Non-Spam	90	510

Predicted

		Spam	Non-Spam
Actual	**Spam**	True Positive	False Negative
	Non-Spam	False Positive	True Negative

Other Metrics

$$\textit{Error Rate} = \frac{False\ Positive + False\ Negative}{Total}$$

$$\textit{False Positive Rate} = \frac{False\ Positive}{False\ Positive + True\ Negative}$$

More Than Two Classes

		Predicted		
	Total = 1000	**A**	**B**	**C**
Actual	**A**	280	8	12
	B	15	260	25
	C	30	50	320

Overall, how often is the model correct?

Metric 1: Accuracy

$$= \frac{280 + 260 + 320}{1000} = 0.86$$

More Than Two Classes

How often does the model correctly identify ~~positives (spam emails)?~~ each class

Metric 2: Recall

(Sensitivity or True Positive Rate)

Predicted

Actual

Total = 1000	A	B	C
A	280	8	12
B	15	260	25
C	30	50	320

$$Recall_A = \frac{280}{280 + 8 + 12} \approx 0.93$$

$$Recall_B = \frac{260}{260 + 15 + 25} \approx 0.87$$

$$Recall_C = \frac{320}{320 + 30 + 50} = 0.8$$

More Than Two Classes

When the model
predicts ~~positive,~~ how
often is it correct?
each class

Metric 3: Precision

		Predicted		
	Total = 1000	A	B	C
Actual	A	280	8	12
	B	15	260	25
	C	30	50	320

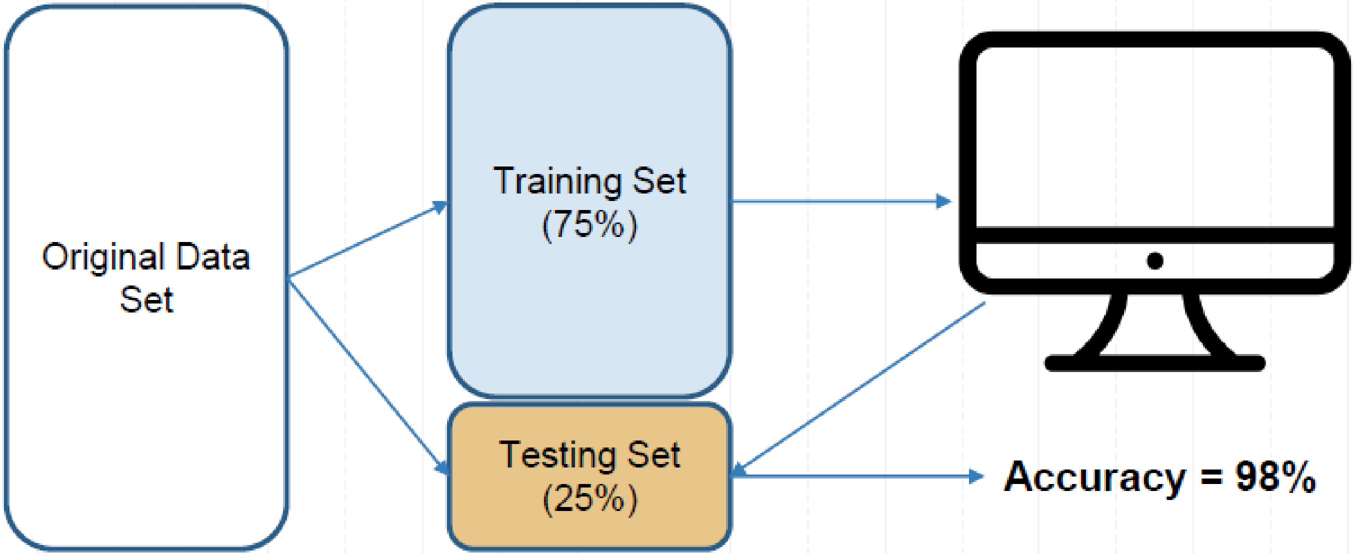
$$Precision_A = \frac{280}{280 + 15 + 30} \approx 0.86$$

$$Precision_B = \frac{260}{260 + 8 + 50} \approx 0.82$$

$$Precision_C = \frac{320}{320 + 12 + 25} \approx 0.90$$

Cross-Validation

Is the Model Ready for Use?



Original Data Set

Training Set (75%)

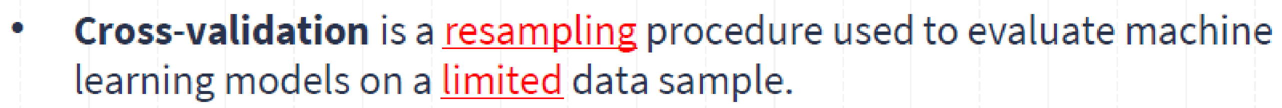
Testing Set (25%)

Accuracy = 98%

What if we split the dataset in a different way?

Should we find another dataset to test our model?

Cross-Validation



- **Cross-validation** is a resampling procedure used to evaluate machine learning models on a limited data sample.

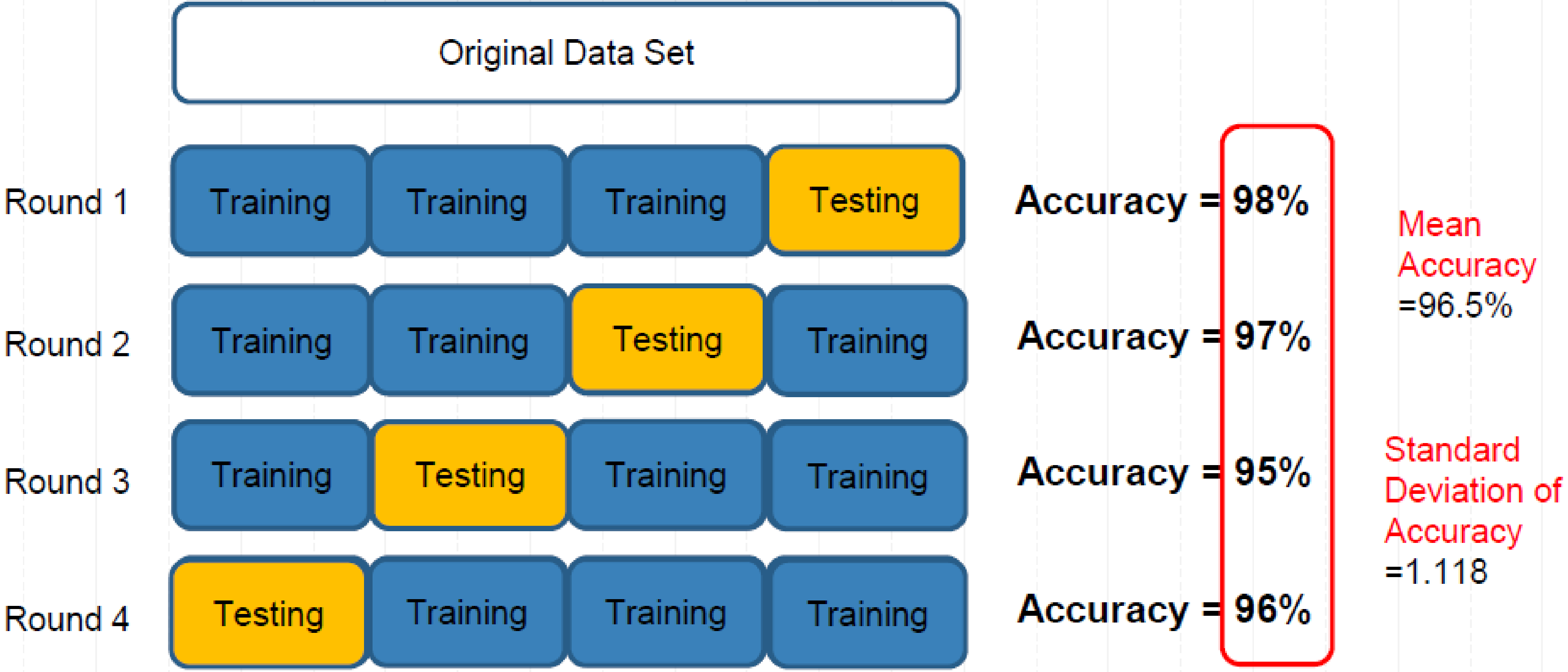
1. Splitting the Data: The dataset is split into K equal-sized (or nearly equal-sized) subsets.

2. Training and Validation: The model is trained K times, each time using K-1 folds for training and the remaining 1 fold for validation.

3. Rotation: The validation fold is rotated such that each of the K folds is used exactly once as the validation set.

4. Averaging the Results: The performance metric (e.g., accuracy, precision, recall) is averaged across all K trials to give an overall performance estimate.

Four-Fold Cross-Validation



Original Data Set

Round 1	Training	Training	Training	Testing	Accuracy = 98%
Round 2	Training	Training	Testing	Training	Accuracy = 97%
Round 3	Training	Testing	Training	Training	Accuracy = 95%
Round 4	Testing	Training	Training	Training	Accuracy = 96%

Mean Accuracy =96.5%

Standard Deviation of Accuracy =1.118

Cross-Validation

- **K**-Fold Cross-validation
- **K** is the number of equal-size blocks you split the data into
- Ten-Fold Cross Validation is a common choice.

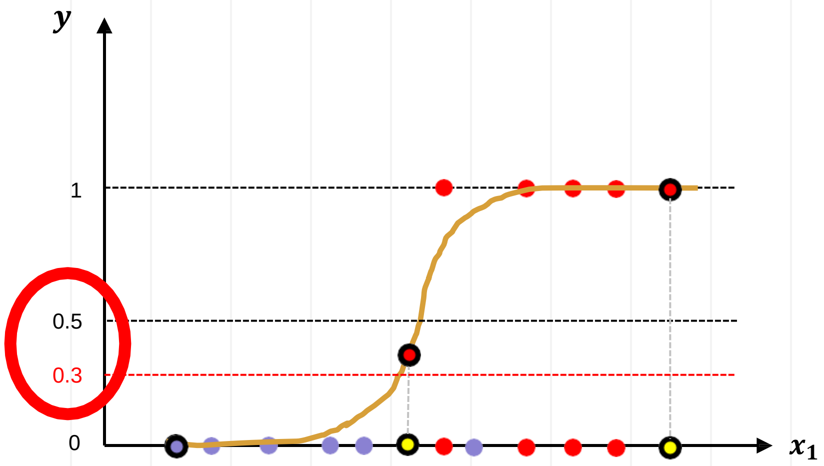
Machine Learning Steps

- Gathering and loading data
- Exploring data (e.g., pandas and visualization)
- Transforming data (e.g., string to numeric)
- Splitting data for training and testing
- Choosing and creating a model
- Training
- Testing (evaluating accuracy)
- Tuning the model (hyperparameters)
- Making predictions on new data

Hyperparameter Tuning

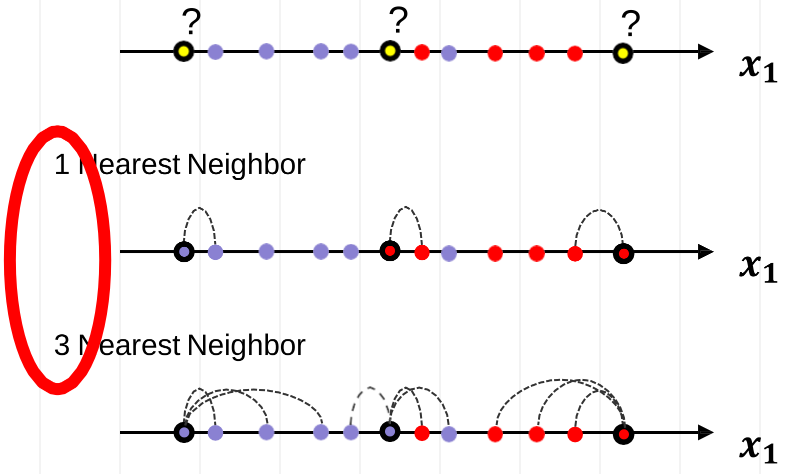
- **Hyperparameter**: model parameters specified in advance (before training)

What **threshold** to use?



Logistic Regression

What **k** to use?



K-Nearest Neighbors

When to Use

- Testing a model
- Hyperparameter Tuning
- Comparing models

Metrics for Accuracy

+

K-Fold Cross-Validation

Machine Learning Steps

- Gathering and loading data
- Exploring data (e.g., pandas and visualization)
- Transforming data (e.g., string to numeric)
- Splitting data for training and testing
- Choosing and creating a model
- Training
- Testing (evaluating accuracy)
- Tuning the model (hyperparameters)
- Making predictions on new data

Python and Machine Learning

5

Python and Machine Learning

Machine Learning

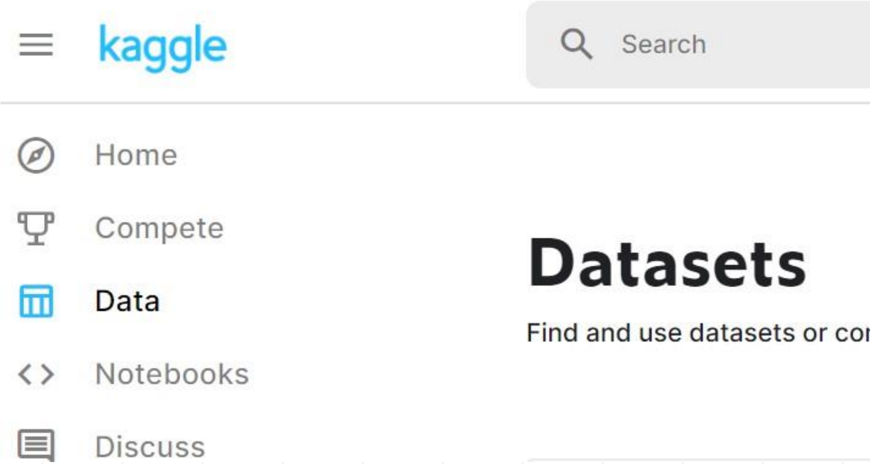
scikit learn

Deep Learning

TensorFlow

K Keras

Machine Learning Datasets



Datasets

Find and use datasets or coı

Machine learning, better, together

https://www.kaggle.com/datasets

https://www.openml.org/

Q&A

Any questions?

You can find me at

wl563@cornell.edu

Machine Learning is Great For

1. Problems for which transitional solutions require a lot of fine-tuning or long lists of rules
2. Complex problems for which the traditional approach yields no good solution
3. Fluctuating environments
4. Getting insights about complex problems and large amounts of data