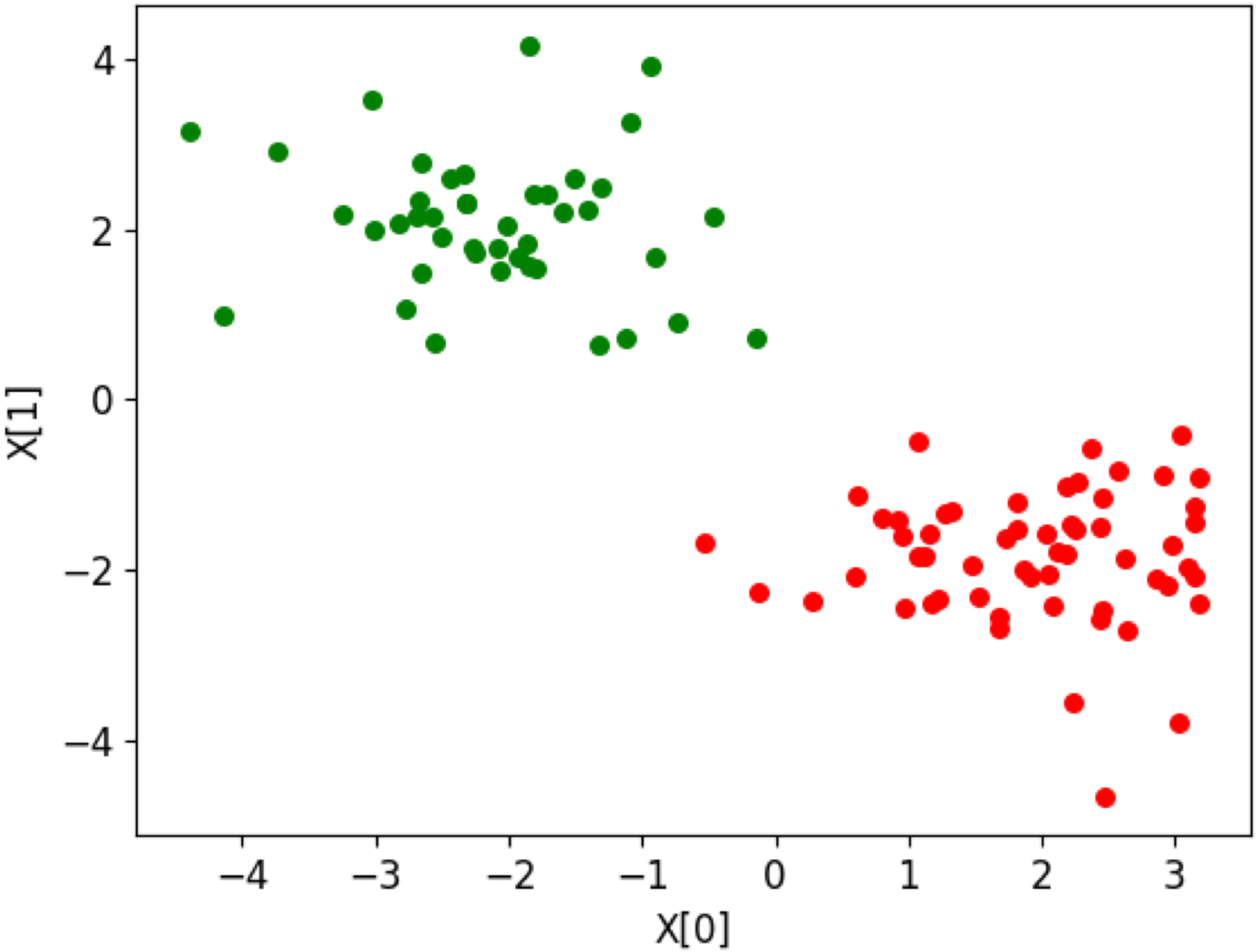
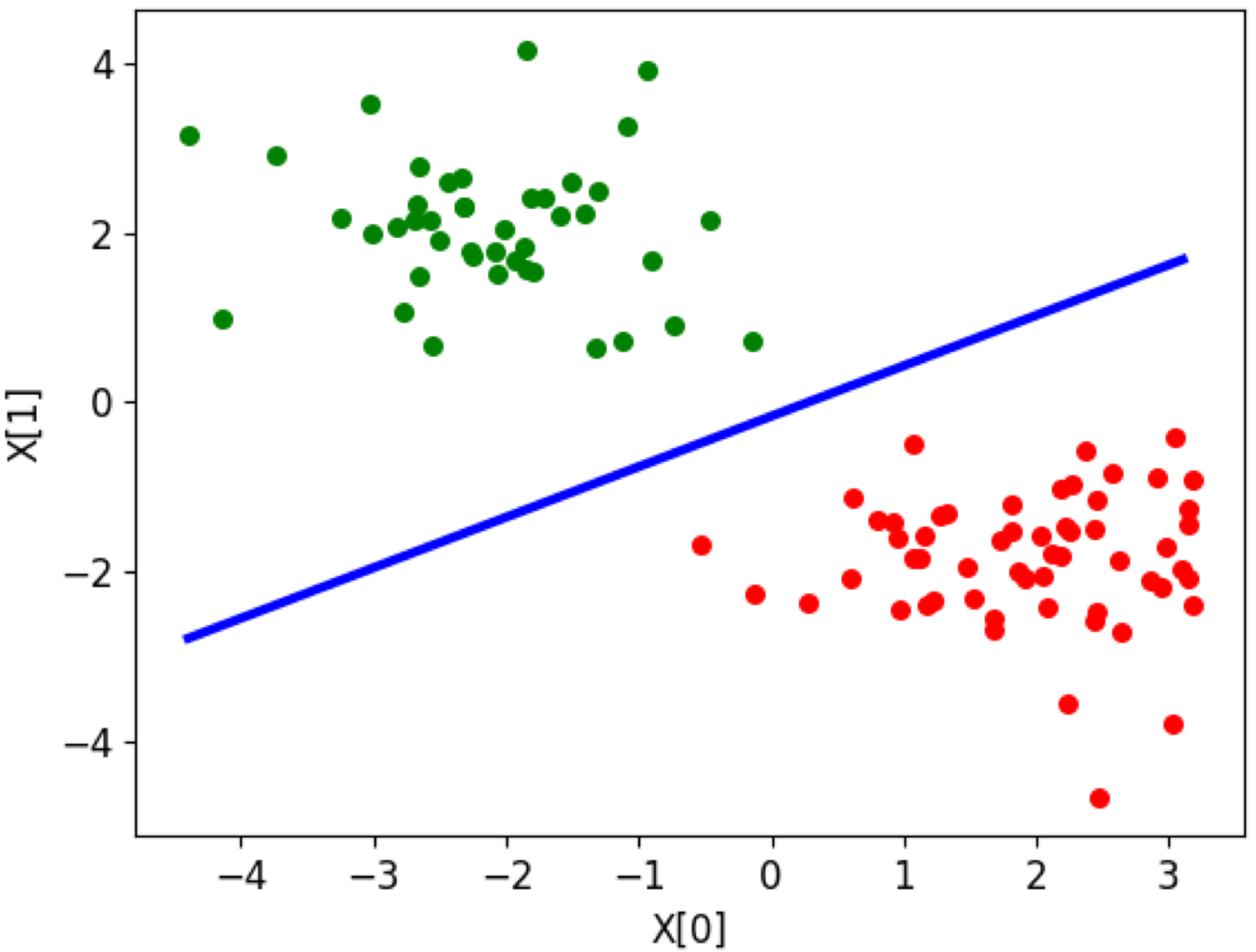
POLITECNICO DI MILANO

Classification

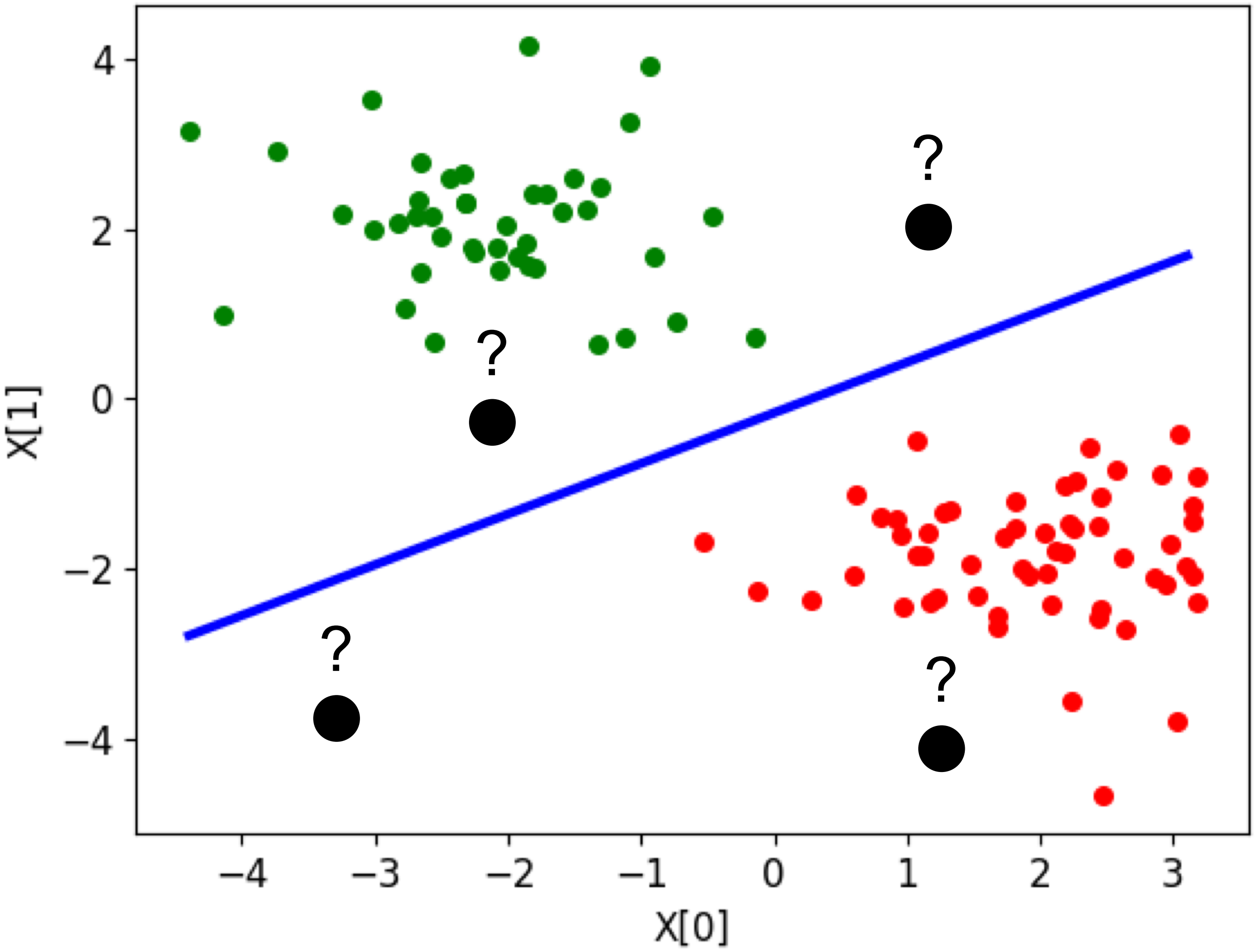
Data Mining and Text Mining (UIC 583 @ Politecnico di Milano)



In classification, we have data that have been labeled according to a certain concept (good/bad, positive/negative, etc.) identified by a supervisor who actually labeled the data



The goal of classification is to compute a model that can discriminate between examples that have been labeled differently, e.g., the points above the planes are green and the one below are red



The model is then used to label previously unseen points.

As for regression, we have to compute a model using known data that will perform well on unknown data.

Contact Lenses Data

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	No	Reduced	None
Young	Myope	No	Normal	Soft
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	No	Reduced	None
Young	Hypermetrope	No	Normal	Soft
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	No	Reduced	None
Pre-presbyopic	Myope	No	Normal	Soft
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	No	Reduced	None
Pre-presbyopic	Hypermetrope	No	Normal	Soft
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	No	Reduced	None
Presbyopic	Myope	No	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	No	Reduced	None
Presbyopic	Hypermetrope	No	Normal	Soft
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

Rows are typically called
examples or data points

Columns are typically called
attributes, variables or features

The target variable to be
predicted is usually called “class”

A Model for the Contact Lenses Data

```
If tear production rate = reduced then recommendation = none
If age = young and astigmatic = no
   and tear production rate = normal then recommendation = soft
If age = pre-presbyopic and astigmatic = no
   and tear production rate = normal then recommendation = soft
If age = presbyopic and spectacle prescription = myope
   and astigmatic = no  then recommendation = none
If spectacle prescription = hypermetrope and astigmatic = no
   and tear production rate = normal then recommendation = soft
If spectacle prescription = myope and astigmatic = yes
   and tear production rate = normal then recommendation = hard
If age young and astigmatic = yes
   and tear production rate = normal then recommendation = hard
If age = pre-presbyopic
   and spectacle prescription = hypermetrope
   and astigmatic = yes then recommendation = none
If age = presbyopic and spectacle prescription = hypermetrope
   and astigmatic = yes then recommendation = none
```

Classification

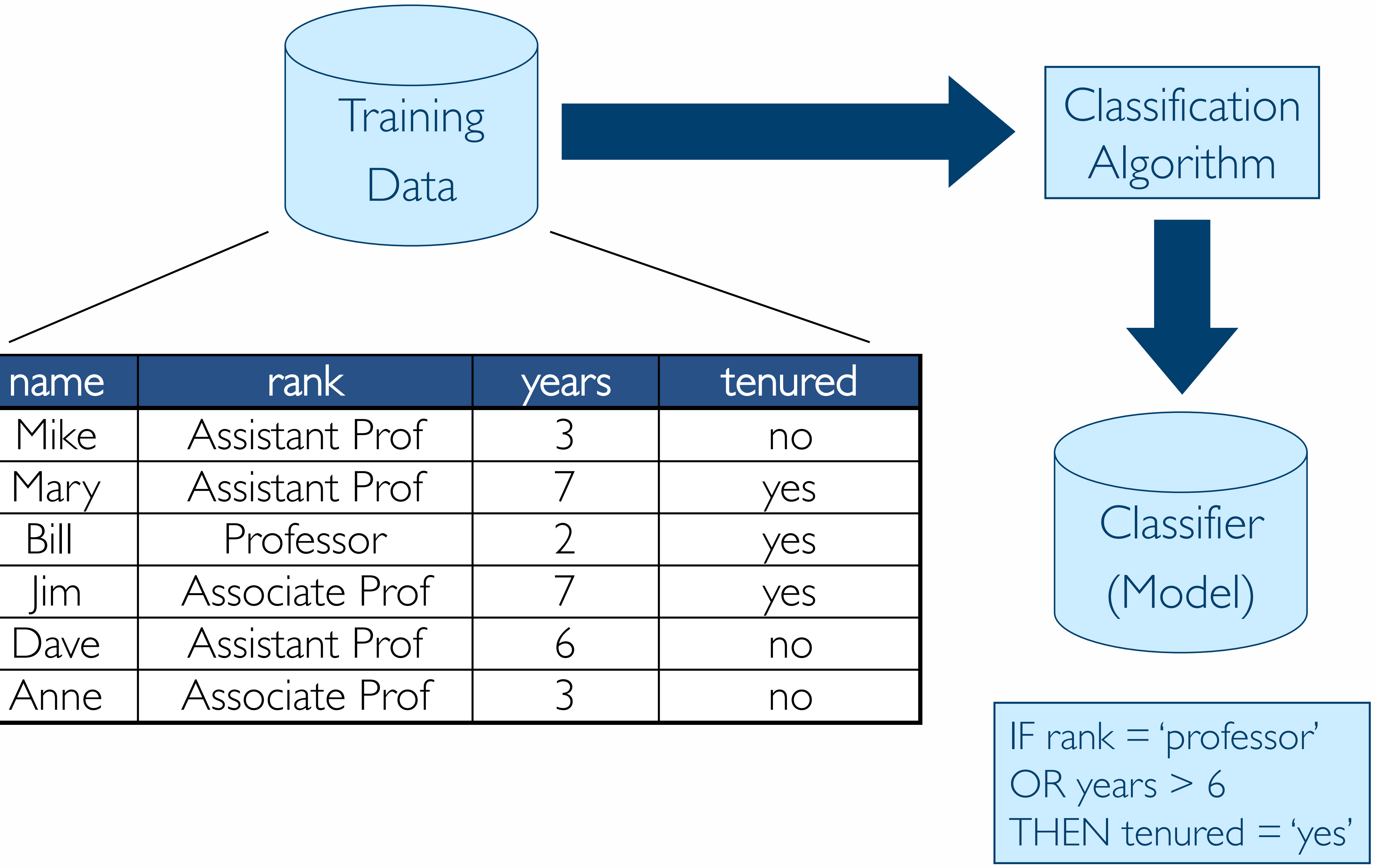
The target to predict is a label (the class variable) (good/bad, none/soft/hard, etc.)

Prediction/Regression

The target to predict is numerical

classification = model building + model usage

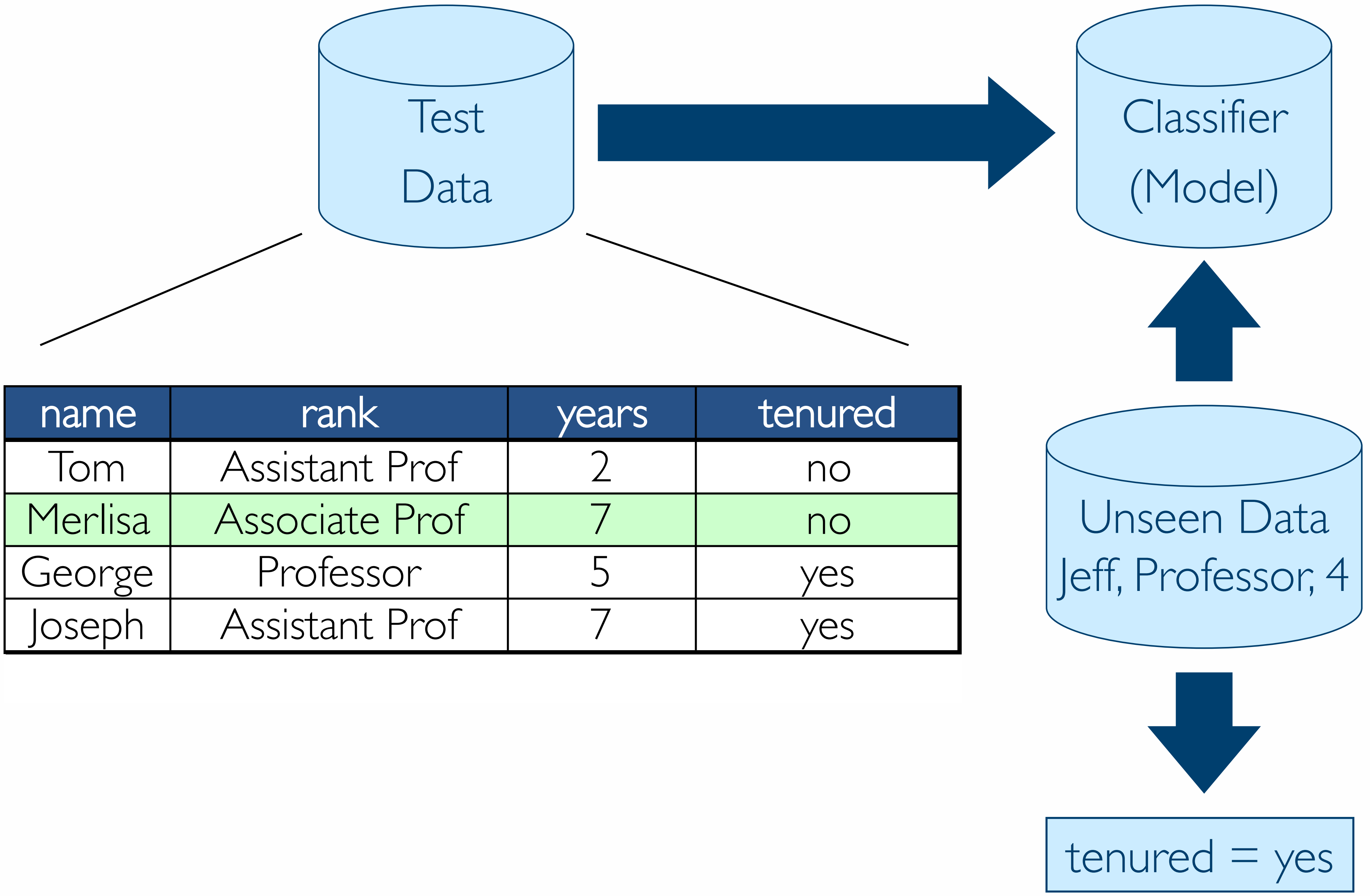
Classification: Model Construction



name	rank	years	tenured
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = 'professor'
OR years > 6
THEN tenured = 'yes'

Classification: Model Usage

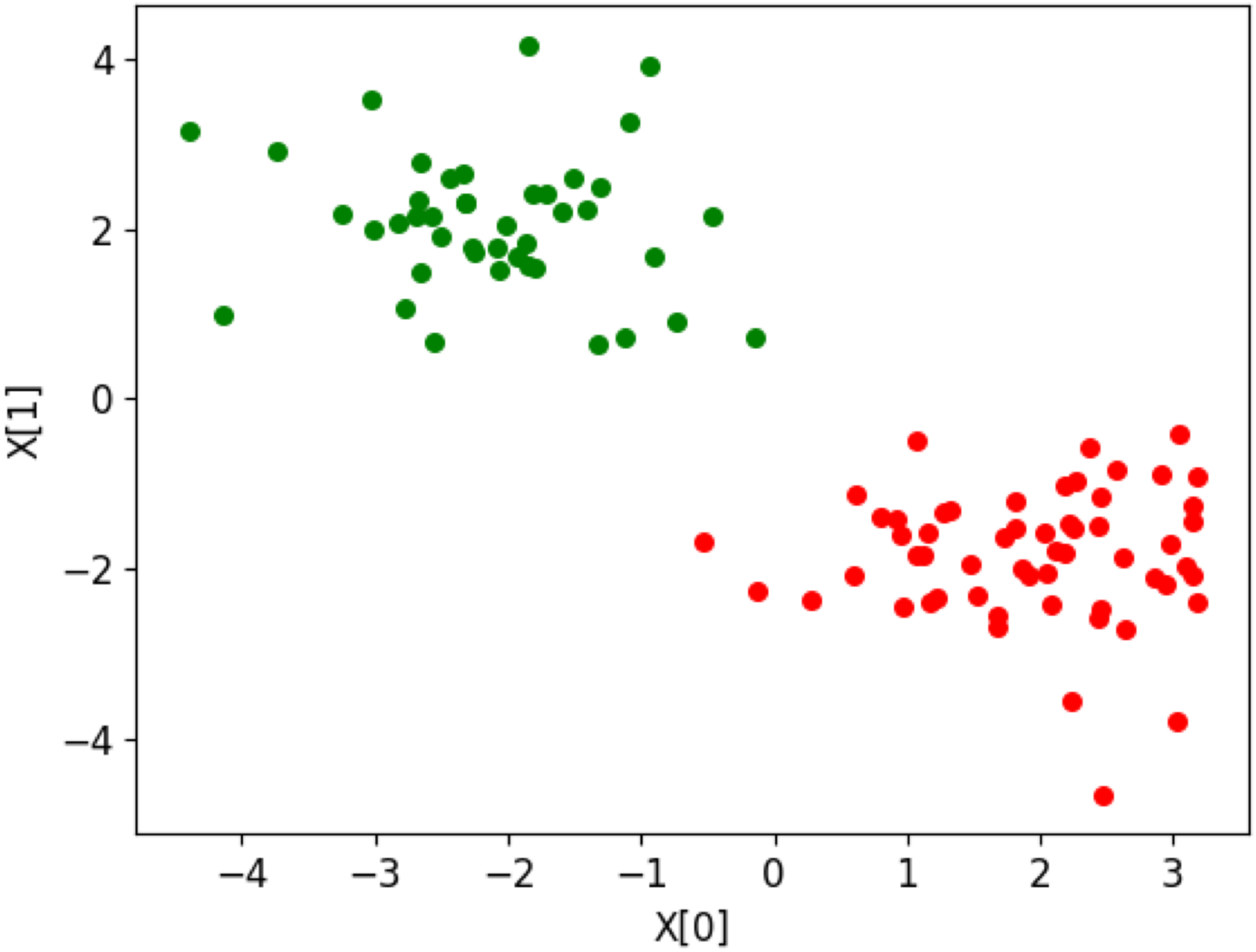


name	rank	years	tenured
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

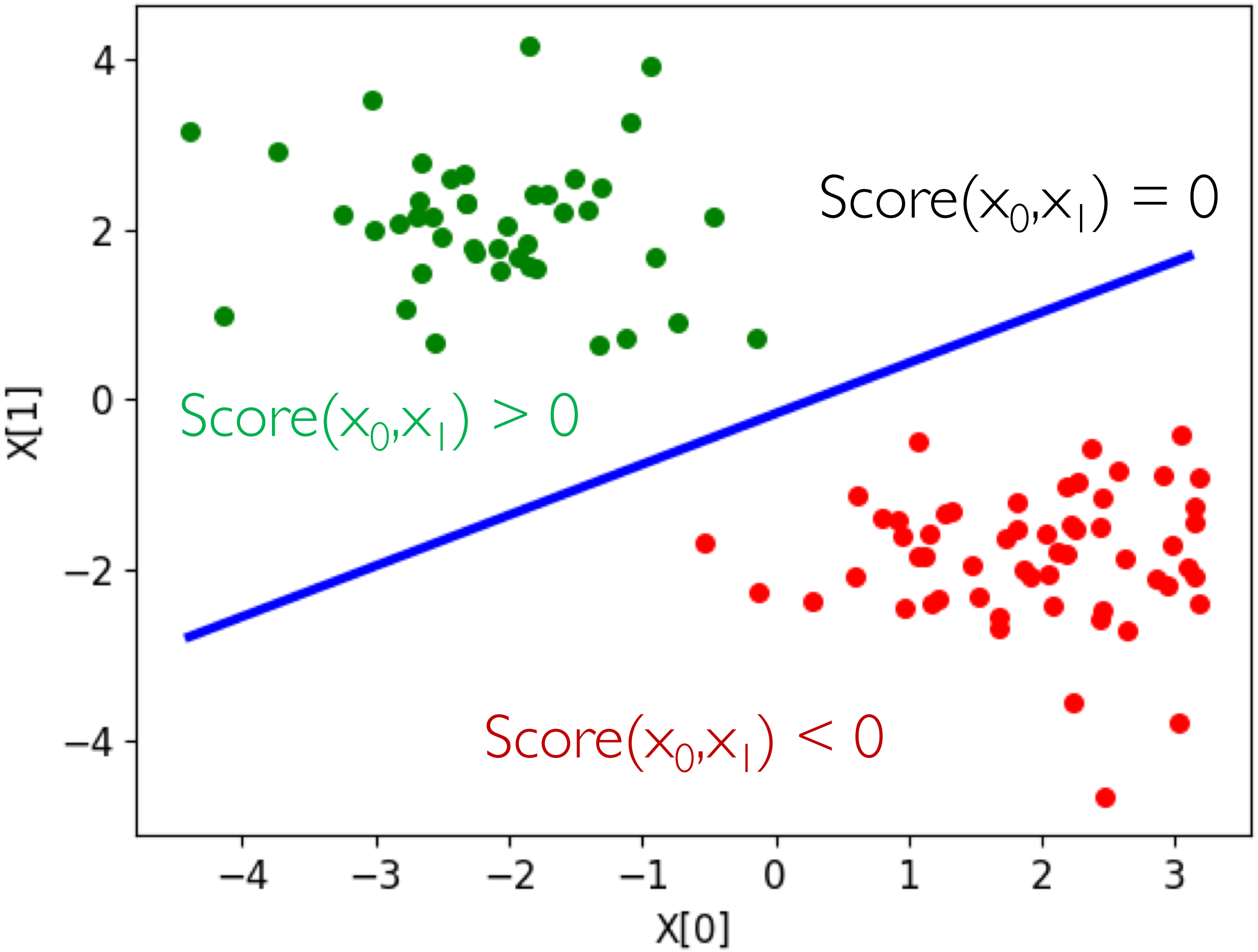
Evaluating Classification Methods

- Accuracy
  - Classifier accuracy in predicting the correct the class labels
- Speed
  - Time to construct the model (training time)
  - Time to use the model to label unseen data
- Other Criteria
  - Robustness in handling noise
  - Scalability
  - Interpretability
  - …

Logistic Regression



To build a model to label the green and red data points we can compute an hyperplane Score(**x**) that separates the examples labeled green from the one labeled red



To predict the labels we check whether the value of Score(x_0, x_1) is positive or negative and we label the examples accordingly

Intuition Behind Linear Classifiers (Two Classes)

- We define a score function similar to the one used in linear regression,

$$Score(\vec{x_i}) = \sum_{j=0}^{D} w_j h_j(\vec{x_i})$$

- The label is determined by the sign of the score value,

$$\hat{y}_i = sign(Score(\vec{x_i})) = \begin{cases} +1 & \texttt{if } Score(\vec{x_i}) \geq 0 \\ -1 & \texttt{if } Score(\vec{x_i}) < 0 \end{cases}$$

Logistic Regression

- Well-known and widely used statistical classification method
- Instead of computing the label, it computes the probability of assigning a class to an example, that is,

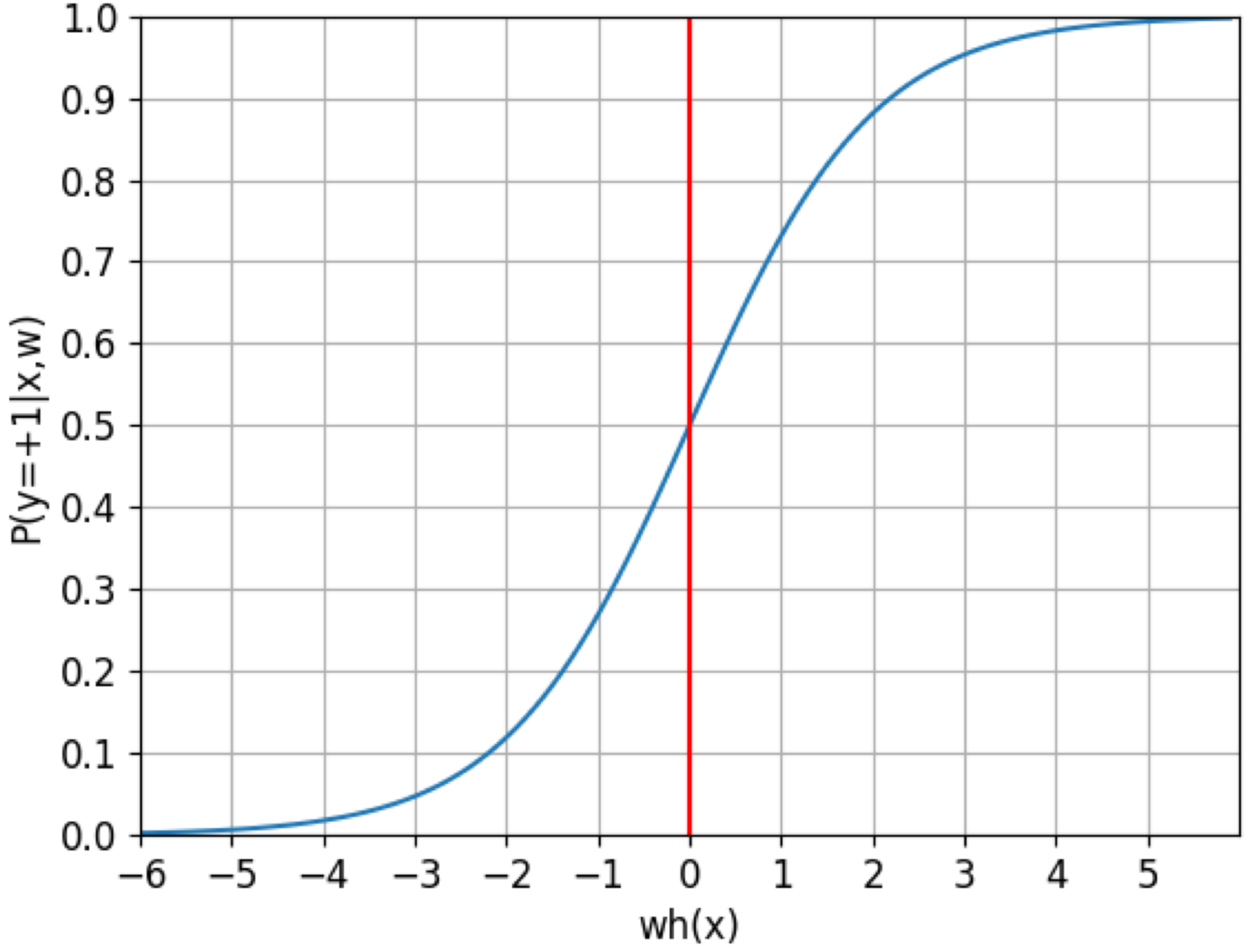
$$P(y_i|\vec{x_i})$$

- For this purpose, logistic regression assumes that,

$$P(\hat{y}_i = +1|\vec{x_i}) = \frac{1}{1+e^{-Score(\vec{x_i})}}$$

- By making the score computation explicit and using h to identify all the feature transformation h_j we get,

$$P(\hat{y}_i = +1|\vec{x_i}, \vec{w}) = \frac{1}{1+e^{-\vec{w}h(\vec{x_i})}}$$



1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
P(y=+1|x,w)
−6
−5
−4
−3
−2
−1
0
1
2
3
4
5
wh(x)

Logistic Regression

- Logistic Regression search for the weight vector that corresponds to the highest likelihood

$$\ell(\vec{w}) = \prod_{i=0}^{N} P(y_i|\vec{x_i}, \vec{w})$$

- For this purpose, it performs a gradient ascent on the log likelihood function

$$\ell\ell(\vec{w}) = \ln \ell(\vec{w})$$

- Which updates weight j using,

$$\frac{\partial \ell\ell}{\partial w_j} = \sum_{I=1}^{N} h_j(\vec{x_i})(1[y_i = +1] - P(y = +1|\vec{x_i}, \vec{w}))$$

Classification using Logistic Regression

- To classify an example X, we select the class Y that maximizes the probability P(Y=y | X) or check

$$\frac{P(y=-1)|x)}{P(y=+1)|x)} > 1 \qquad \text{or} \qquad e^{-\vec{w}h(\vec{x})} > 1$$

- By taking the natural logarithm of both sides, we obtain a linear classification rule that assign the label Y=-1 if

$$\sum_{i=0}^{D} w_i h_i(x) < 0$$

- Y=+1 otherwise

Overfitting & Regularization

L_1 & L_2 Regularization

- Logistic regression can use L_1 and L_2 regularization to limit overfitting and produce sparse solution
- Like it happened with regression, overfitting is often associated to large weights so to limit overfitting we penalize large weights

Total cost = Measure of Fit - Magnitude of Coefficients

- Regularization
  - L_1 uses the sum of absolute values, which penalizes large weights and at the same time promotes sparse solutions
  - L_2 uses the sum of squares, which penalizes large weights

L_1 Regularization

$$\ell(\vec{w}) - \alpha||\vec{w}||_1$$

L_2 Regularization

$$\ell(\vec{w}) - \alpha||\vec{w}||_2^2$$

If α is zero, then we have no regularization

If α tends to infinity,
the solution is a zero weight vector

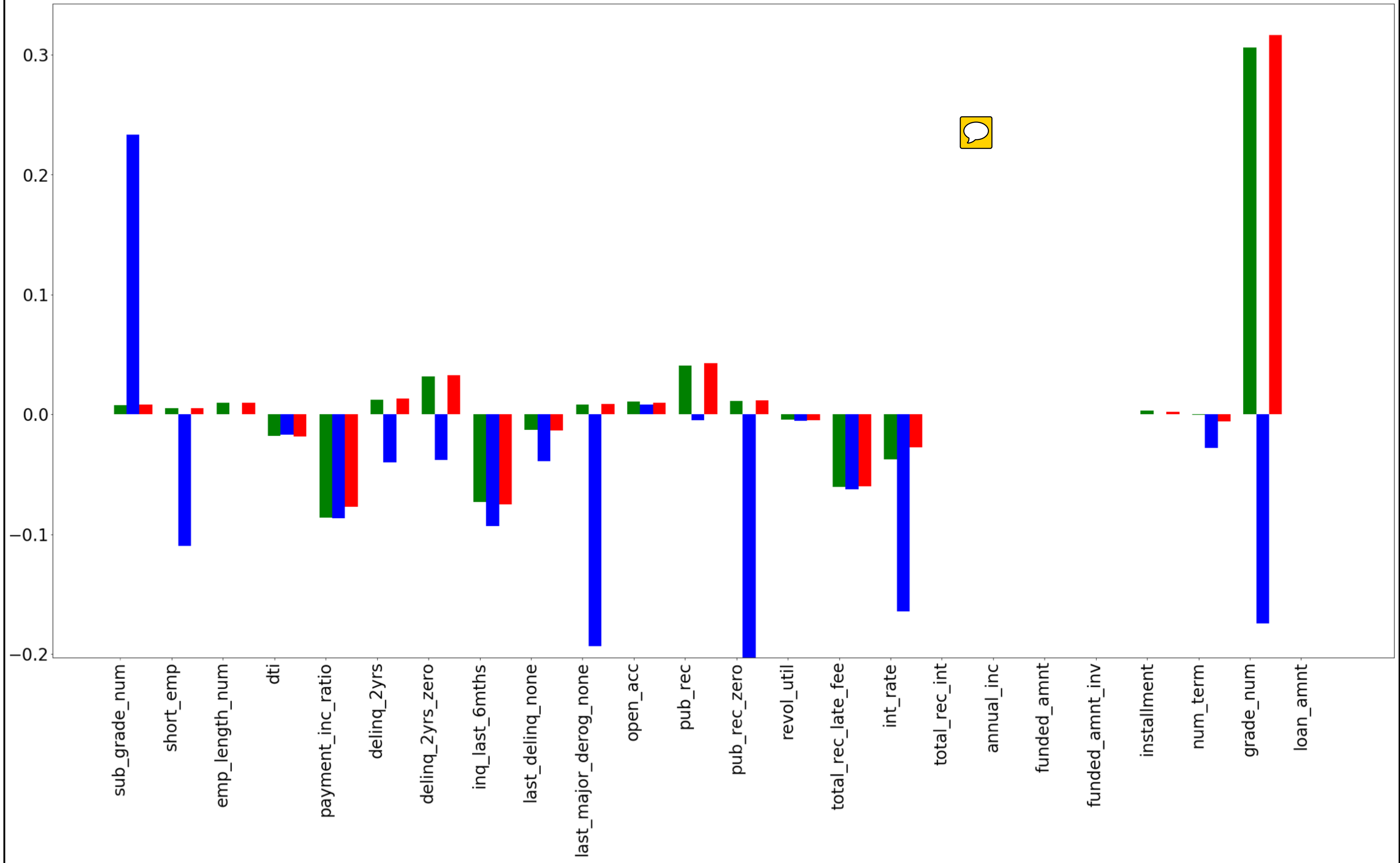
α must balance fit and the weight magnitude

Example

Safe Loans Prediction

- Given data downloaded from the Lending Club Corporation website regarding safe and risky loans, build a model to predict whether a loan should be given

- Model evaluation using 10-fold crossvalidation
  - Simple Logistic Regression μ=0.63 σ=0.05
  - Logistic Regression (L1) μ=0.63 σ=0.07
  - Logistic Regression (L2) μ=0.64 σ=0.05



Weights of the simple logistic model, the L1 regularized model, and the L2 regularized model.

From now on we are going to use k-fold crossvalidation a lot to score models

k-fold crossvalidation generates k separate models

Which one should be deployed?

K-fold crossvalidation provides an evaluation of model performance on unknown data

Its output it’s the evaluation, not the model!

The model should be obtained by running the algorithm on the entire dataset!

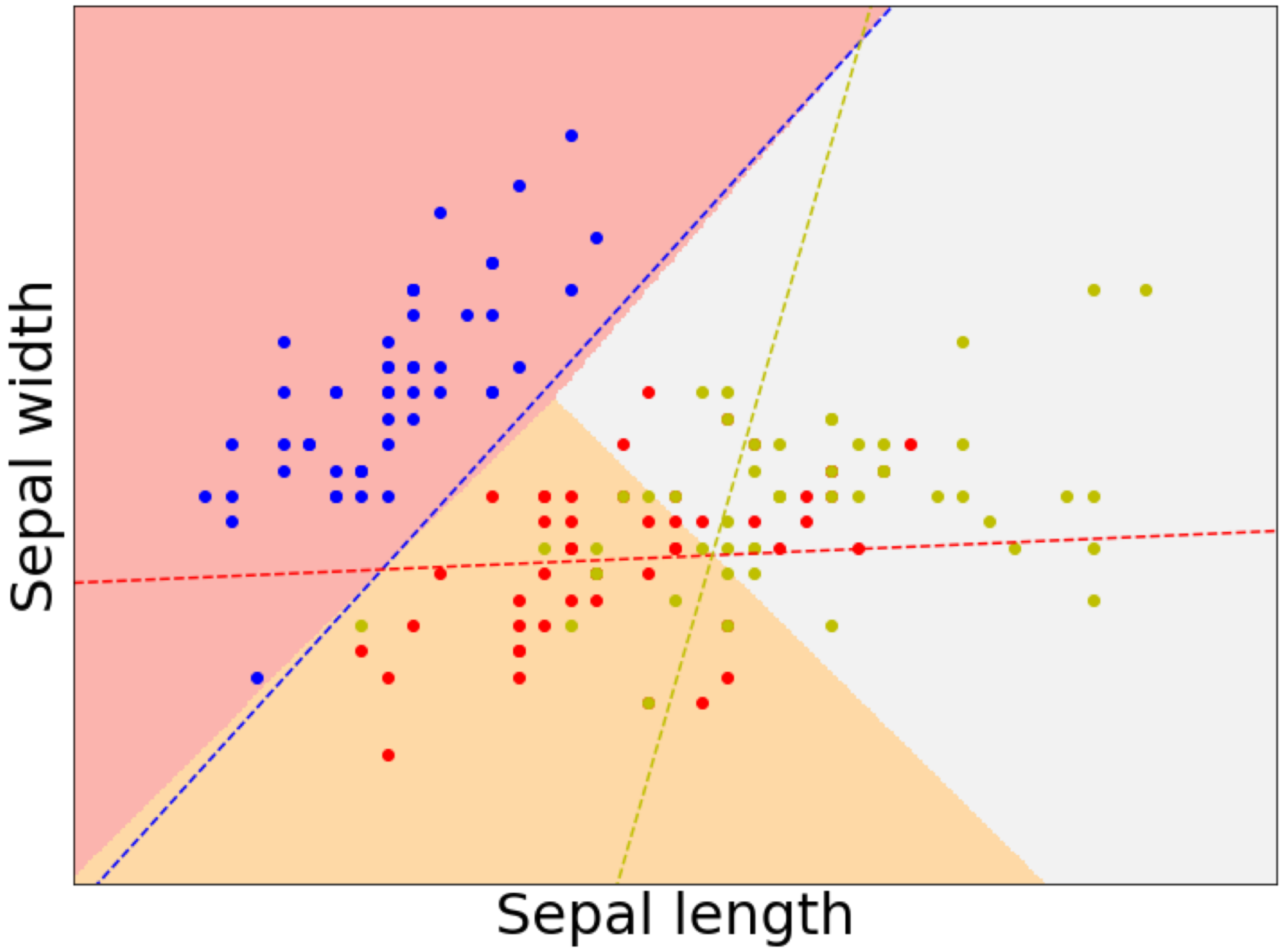
Multiclass Classification

Logistic regression assumes that
there are only two class values (e.g., +1/-1)

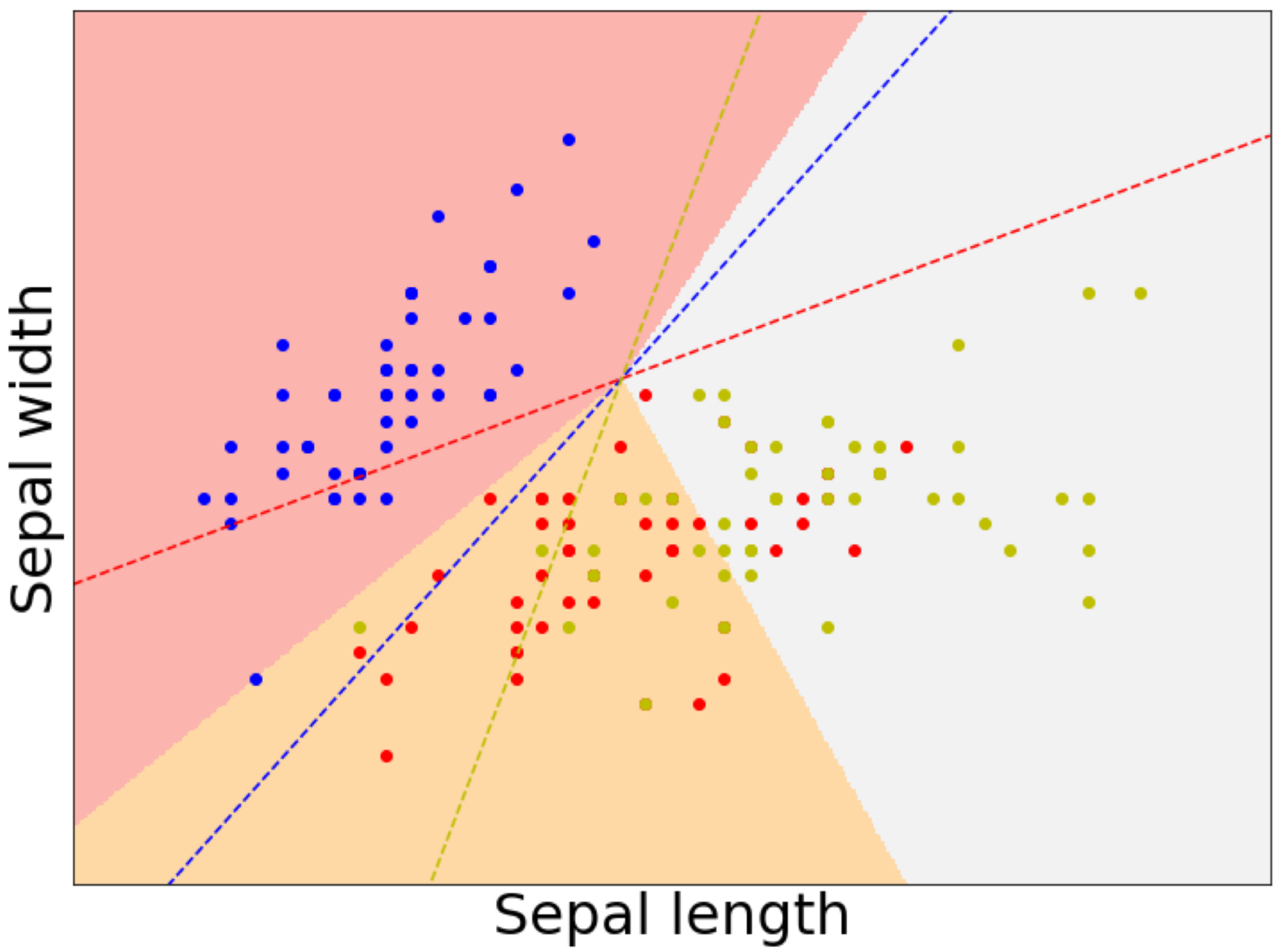
What if we have more?
(e.g., none, soft, hard or Setosa/Versicolour/Virginica)

One Versus the Rest Evaluation

- For each class, it creates one classifier that predicts the target class against all the others
- For instance, given three classes A, B, C, it computes three models: one that predicts A against B and C, one that predicts B against A and C, and one that predicts C against A and B
- Then, given an example, all the three classifiers are applied and the label with the highest probability is returned
- Alternative approaches include the minimization of loss based on the multinomial loss fit across the entire probability distribution



One Versus Rest multiclass model using the Iris dataset



Multinomial multiclass model using the Iris dataset

Run the Logistic Regression Python notebooks on multiclass classification available on Beep

Categorical Attributes

So far we applied Logistic Regression (and regression) to numerical variables

What if the variables are categorical?

The Weather Dataset

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

One Hot Encoding

Map each categorical attribute with n values into n binary 0/1 variables

Each one describing one specific attribute values

For example, attribute Outlook is replaced by three binary variables Sunny, Overcast, and Rainy

overcast	rainy	sunny
0	0	1
0	0	1
1	0	0
0	1	0
0	1	0
0	1	0
1	0	0
0	0	1
0	0	1
0	1	0
0	0	1
1	0	0
1	0	0
0	1	0

One Hot Encoding for the Outlook attribute.

Run the Logistic Regression Python notebooks provided on Beep

Summary

Classification

- The goal of classification is to build a model of a target concept (the class attribute) available in the data
- The target concept is typically a symbolic label (good/bad, yes/no, none/soft/hard, etc.)
- Linear classifiers are one of the simplest of classification algorithm that computes a model represented as a hyperplane that separates examples of the two classes
- Logistic regression is a classification algorithm that computes the probability to classify an example with a certain label
- It works similarly to linear regression and can be coupled with L1 and L2 regularization to limit overfitting
- It assumes that the class has only two values but it can be extended to problems involving more than two classes
- Like linear regression it assumes that the variables are numerical, if they are not, categorical variables must be mapped into numerical values
- One of the typical approaches is to use one-hot-encoding