Feature Normalization: The test set must use identical scaling to the training set

- Fit the scaler using the training set, then apply the same scaler to transform the test set.
- Do not scale the training and test sets using different scalers: this could lead to random skew in the data.
- Do not fit the scaler using any part of the test data: referencing the test data can lead to a form of *data leakage*. More on this issue later in the course.

Ridge Regression

- Ridge regression learns *w*, *b* using the same least-squares criterion but adds a penalty for large variations in *w* parameters

$$RSS_{RIDGE}(\boldsymbol{w}, b) = \sum_{\{i=1\}}^{N} (\boldsymbol{y}_i - (\boldsymbol{w} \cdot \boldsymbol{x}_i + b))^2 \boxed{+ \alpha \sum_{\{j=1\}}^{p} w_j^2}$$

- Once the parameters are learned, the ridge regression <u>prediction</u> formula is the <u>same</u> as ordinary least-squares.
- The addition of a parameter penalty is called <u>regularization</u>. Regularization prevents overfitting by restricting the model, typically to reduce its complexity.
- Ridge regression uses <u>L2 regularization</u>: minimize sum of squares of *w* entries
- The influence of the regularization term is controlled by the α parameter.
- Higher alpha means more regularization and simpler models.

Lasso regression is another form of regularized linear regression that uses an L1 regularization penalty for training (instead of ridge's L2 penalty)

- L1 penalty: Minimize the sum of the absolute values of the coefficients

$$RSS_{LASSO}(\boldsymbol{w}, b) = \sum_{\{i=1\}}^{N} (\boldsymbol{y}_i - (\boldsymbol{w} \cdot \boldsymbol{x}_i + b))^2 \boxed{+ \alpha \sum_{\{j=1\}}^{p} |w_j|}$$

- This has the effect of setting parameter weights in *w* to zero for the least influential variables. This is called a sparse solution: a kind of feature selection
- The parameter α controls amount of L1 regularization (default = 1.0).
- The prediction formula is the same as ordinary least-squares.
- When to use ridge vs lasso regression:
  - *Many small/medium sized effects: use ridge.*
  - *Only a few variables with medium/large effect: use lasso.*

Polynomial Features with Linear Regression

$$\boldsymbol{x} = (x_0, x_1) \Rightarrow \boldsymbol{x}' = (x_0, x_1, x_0^2, x_0 x_1, x_1^2)$$

$$\hat{y} = \hat{w}_0 x_0 + \hat{w}_1 x_1 + \hat{w}_{00} x_0^2 + \hat{w}_{01} x_0 x_1 + \hat{w}_{11} x_1^2 + b$$

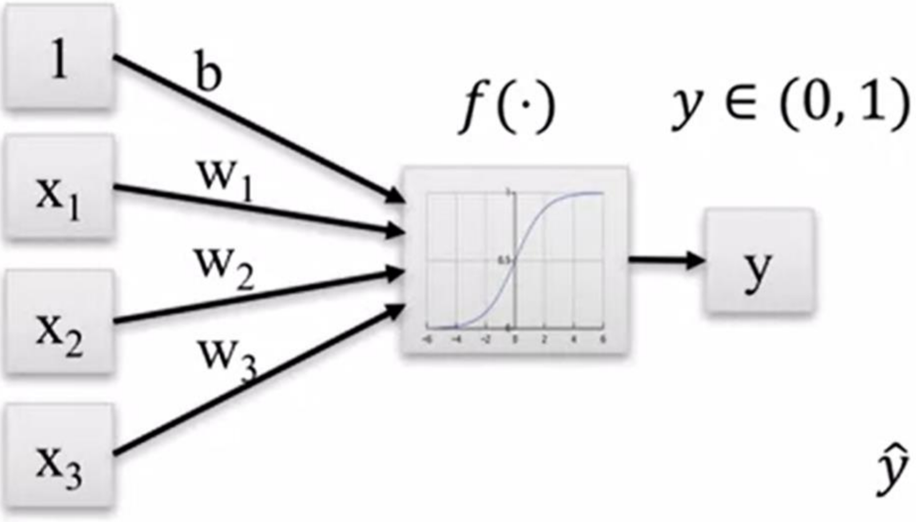
- **Generate new features consisting of all polynomial combinations of the original two features (x_0, x_1).**
- **The *degree* of the polynomial specifies how many variables participate at a time in each new feature (above example: degree 2)**
- **This is still a weighted linear combination of features, so it's <u>still a linear model</u>, and can use same least-squares estimation method for *w* and *b*.**

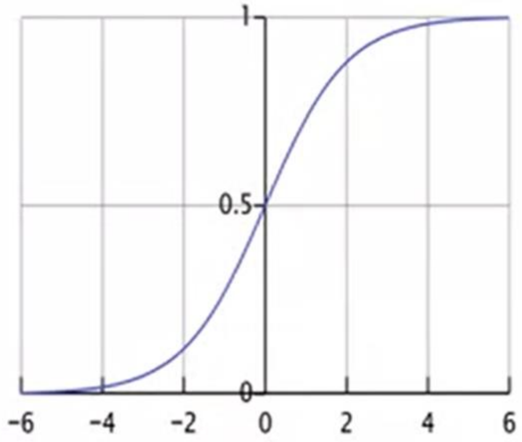
Polynomial Features with Linear Regression

- **Why would we want to transform our data this way?**
  - *To capture interactions between the original features by adding them as features to the linear model.*
  - *To make a classification problem easier (we'll see this later).*
- **More generally, we can apply other non-linear transformations to create new features**
  - *(Technically, these are called non-linear basis functions)*
- **Beware of polynomial feature expansion with high degree, as this can lead to complex models that overfit**
  - *Thus, polynomial feature expansion is often combined with a regularized learning method like ridge regression.*

Linear models for classification: Logistic Regression

Input features

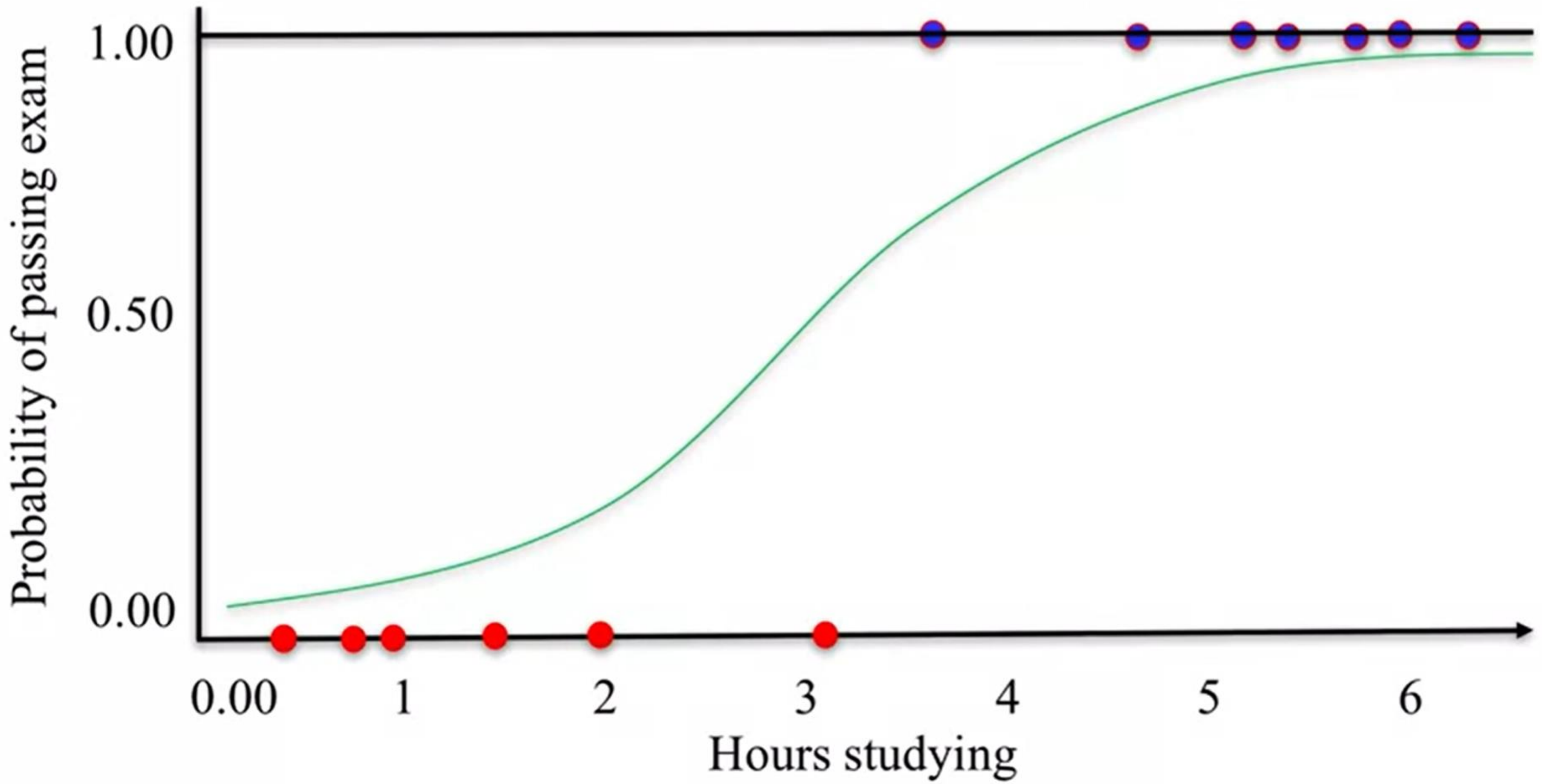




$$\hat{y} = \text{logistic}(\hat{b} + \hat{w}_1 \cdot x_1 + \cdots \hat{w}_n \cdot x_n)$$

$$= \frac{1}{1 + \exp\left[-\left(\hat{b} + \hat{w}_1 \cdot x_1 + \cdots \hat{w}_n \cdot x_n\right)\right]}$$

Linear models for classification: Logistic Regression



Linear classifiers: how would you separate these two groups of training examples with a straight line?

Feature vector

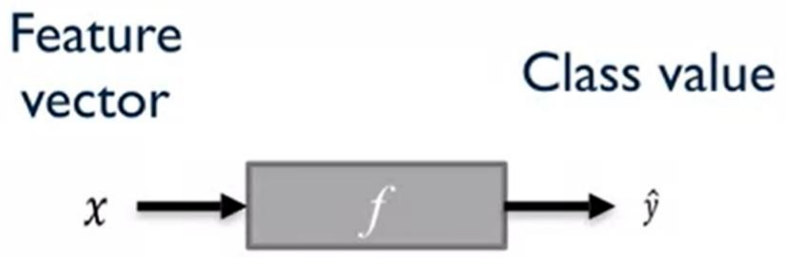
Class value

$$x \longrightarrow \boxed{f} \longrightarrow \hat{y}$$

$$f(x, w, b) = sign(w \circ x + b)$$

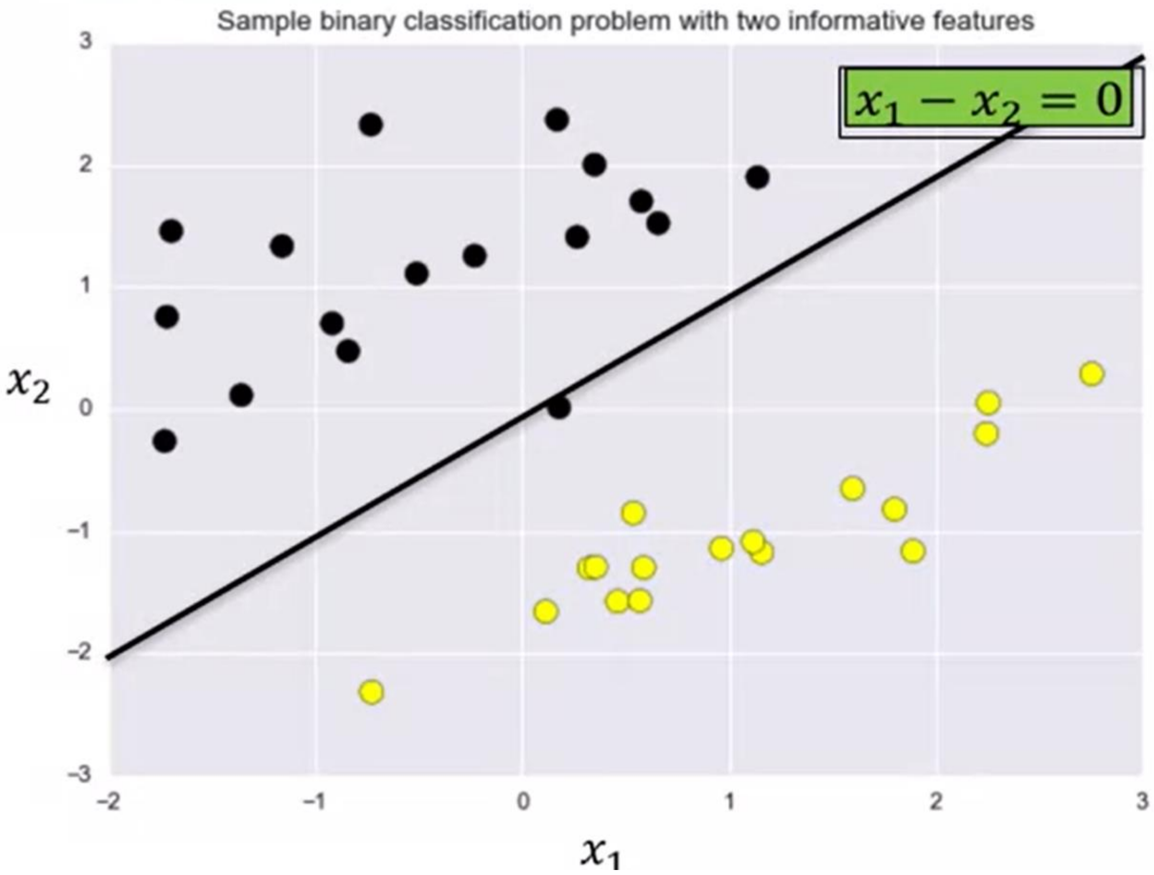
$$= sign\left(\sum w[i]x[i] + \mathrm{b}\right)$$

Linear classifiers: how would you separate these two groups of training examples with a line?



$$f(x, w, b) = sign(w \circ x + b)$$

$$x_1 - x_2 = 0$$
$$w = [1, -1]$$
$$b = 0$$

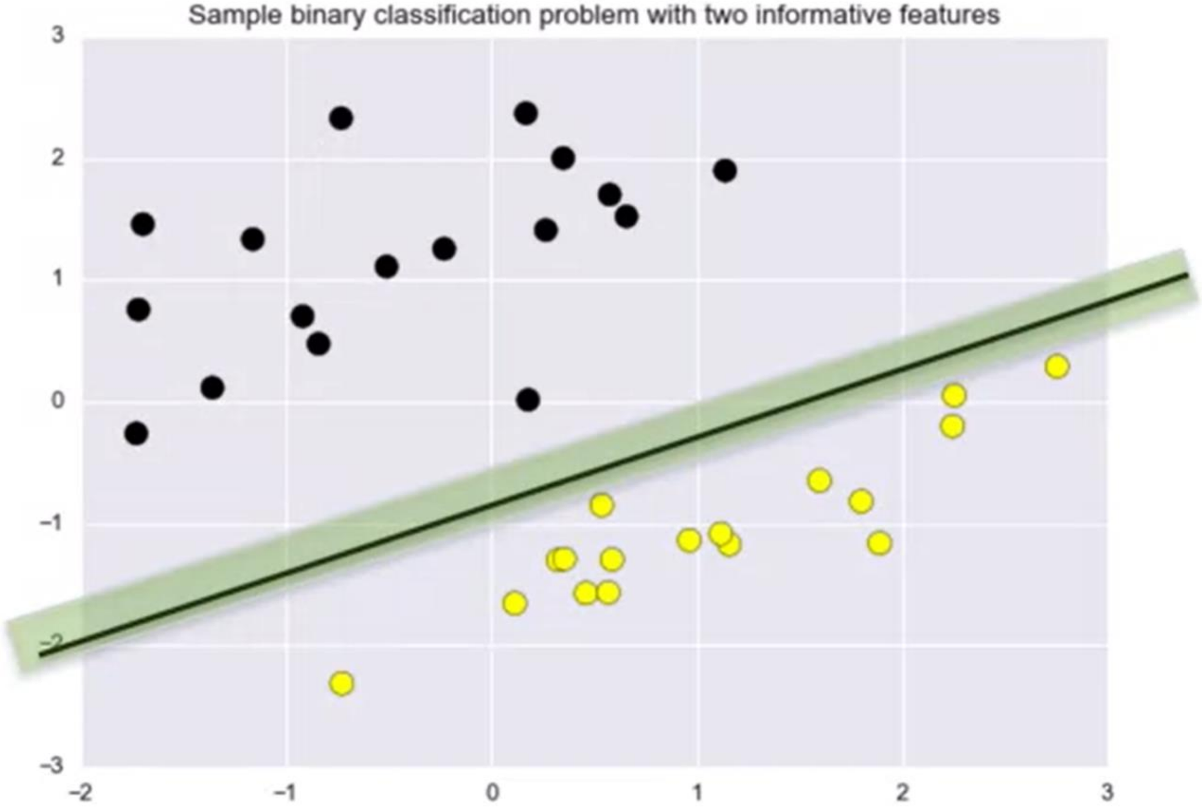


Classifier Margin

$$f(x, w, b) = sign(w \circ x + b)$$

Classifier margin

Defined as the maximum width the decision boundary area can be increased before hitting a data point.

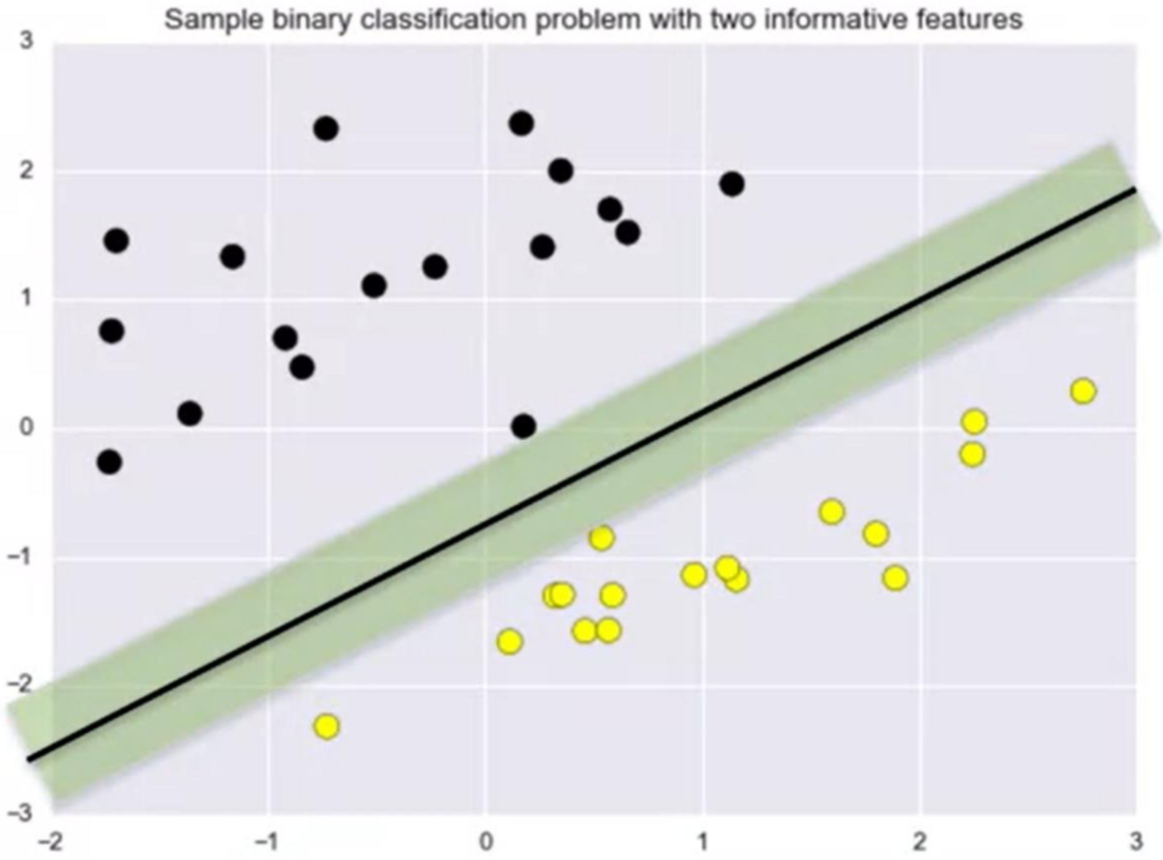


Maximum Margin Linear Classifier: Linear Support Vector Machines

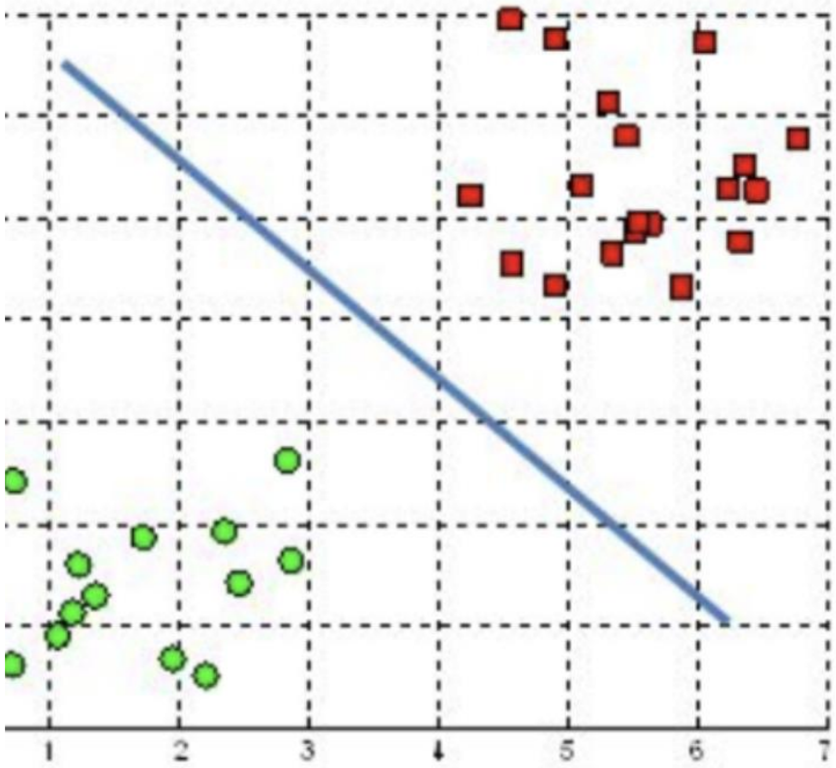
$$f(x, w, b) = sign(w \circ x + b)$$

<u>Maximum margin classifier</u>

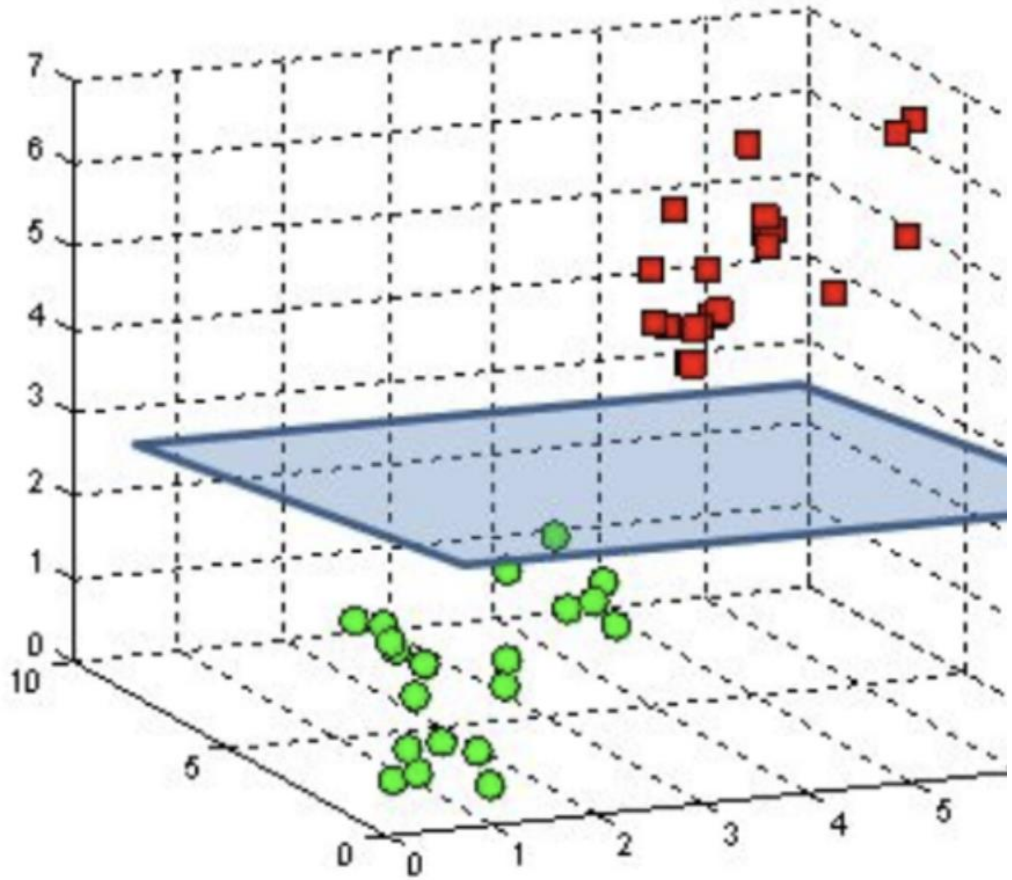
The linear classifier with maximum margin is a linear Support Vector Machine (LSVM).



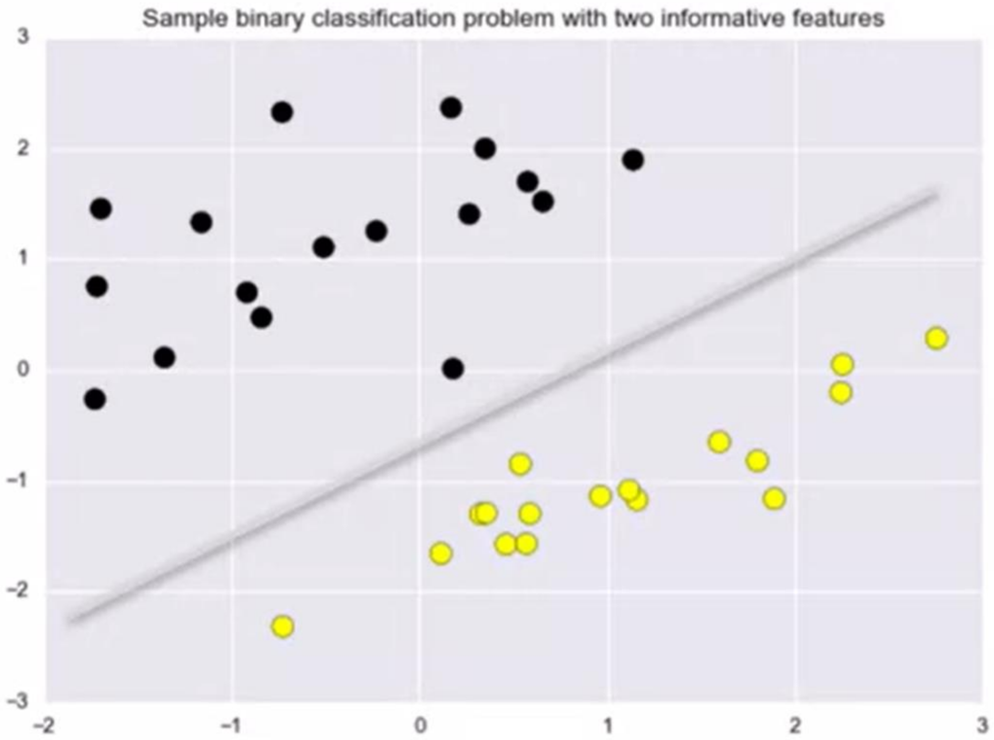
yperplane in $\mathbb{R}^2$ is a line



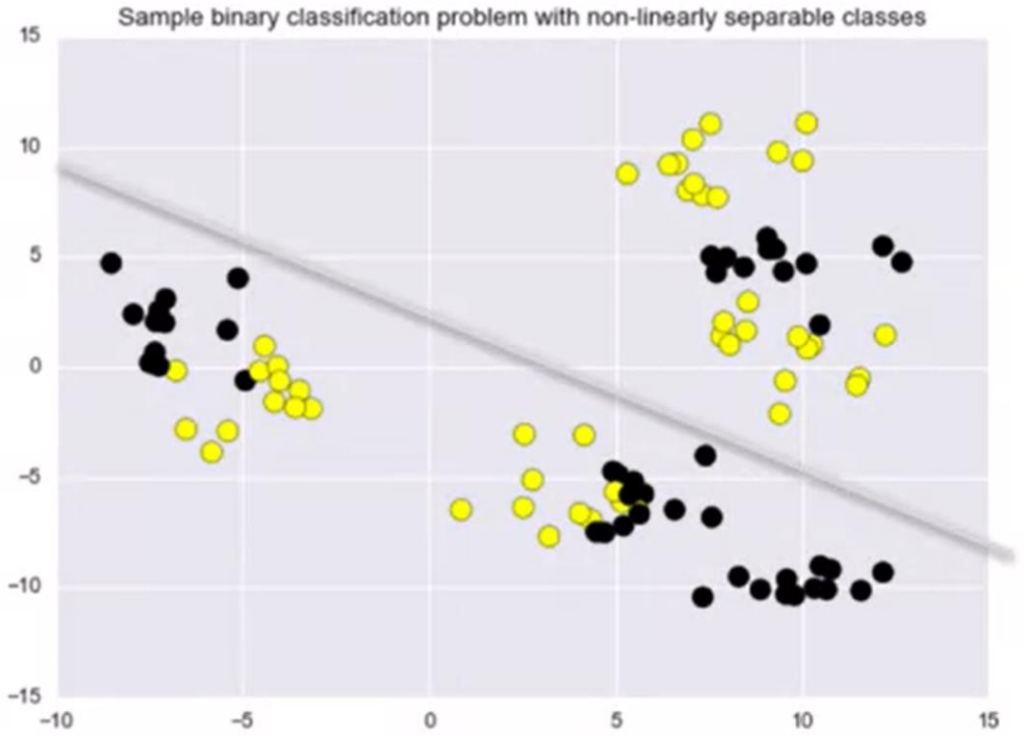
A hyperplane in $\mathbb{R}^3$ is a plane



But what about more complex binary classification problems?



Easy for a linear classifier



Difficult/impossible for a linear classifier