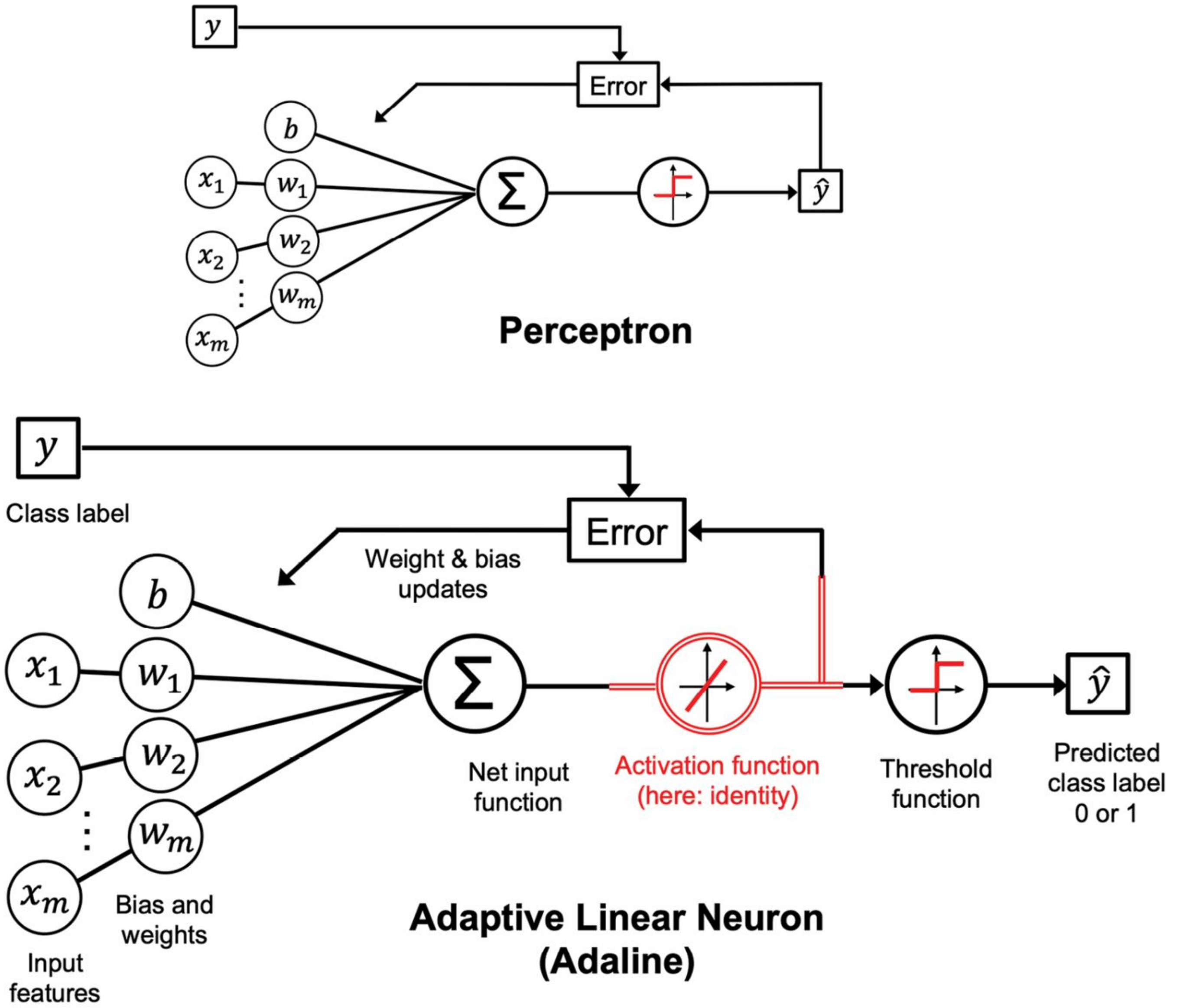
THE UNIVERSITY of TENNESSEE KNOXVILLE

Lecture 07: Linear and Logistic Regression

Sergei V. Kalinin

Perceptron and Adaline



From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

Training Linear Neuron

- Initialize the weights and bias unit to 0 or small random numbers
- For each training example, ***x(i)***:
- Compute the output value, $\boldsymbol{y(i) = w^T x(i) + b}$
- Update the weights and bias unit: $w_j := w_j + \Delta w_j$ and $b := b + \Delta b$
- Where $\Delta w_j = \eta\left(y^{(i)} - \hat{y}^{(i)}\right)x_j^{(i)}$ and $\Delta b = \eta\left(y^{(i)} - \hat{y}^{(i)}\right)$

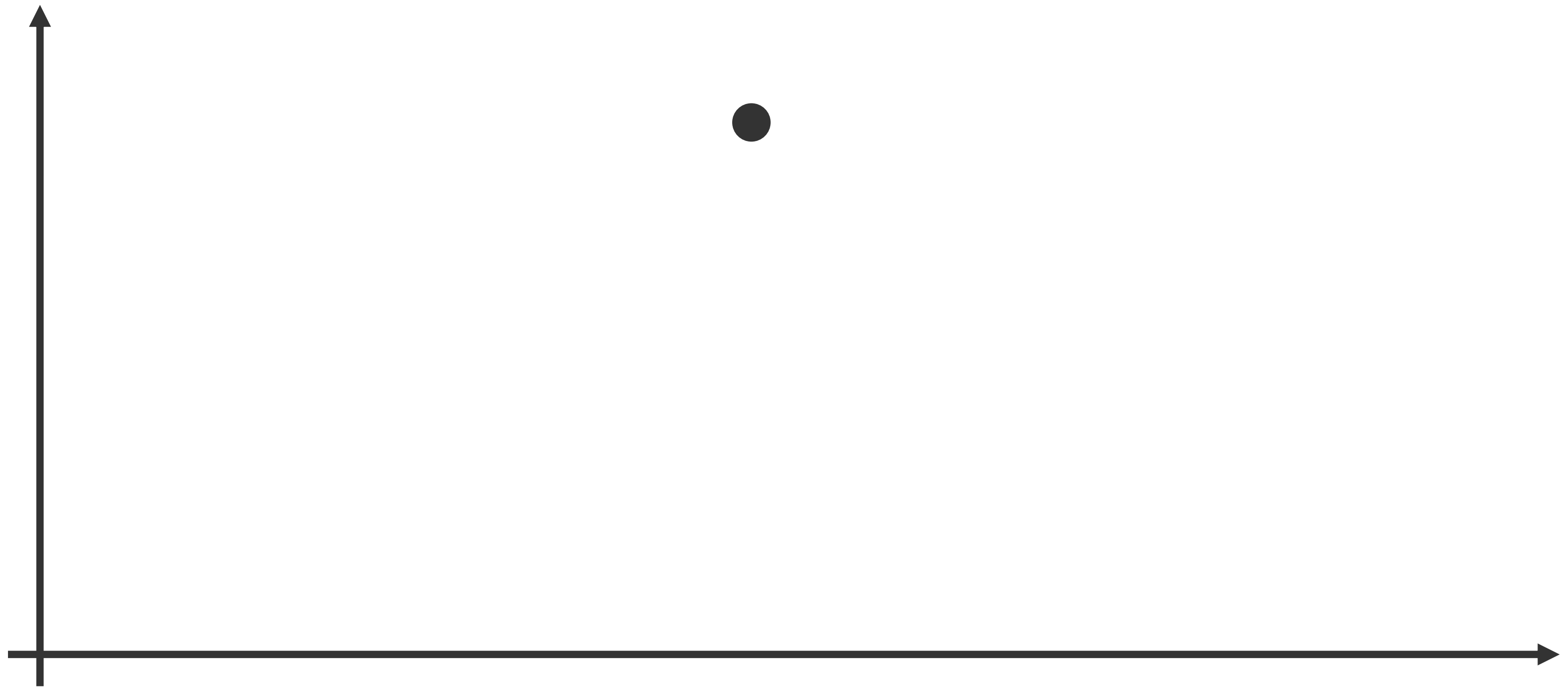
Each weight, w_j, corresponds to a feature, x_j, in the dataset,

η is the **learning rate** (typically a constant between 0.0 and 1.0),

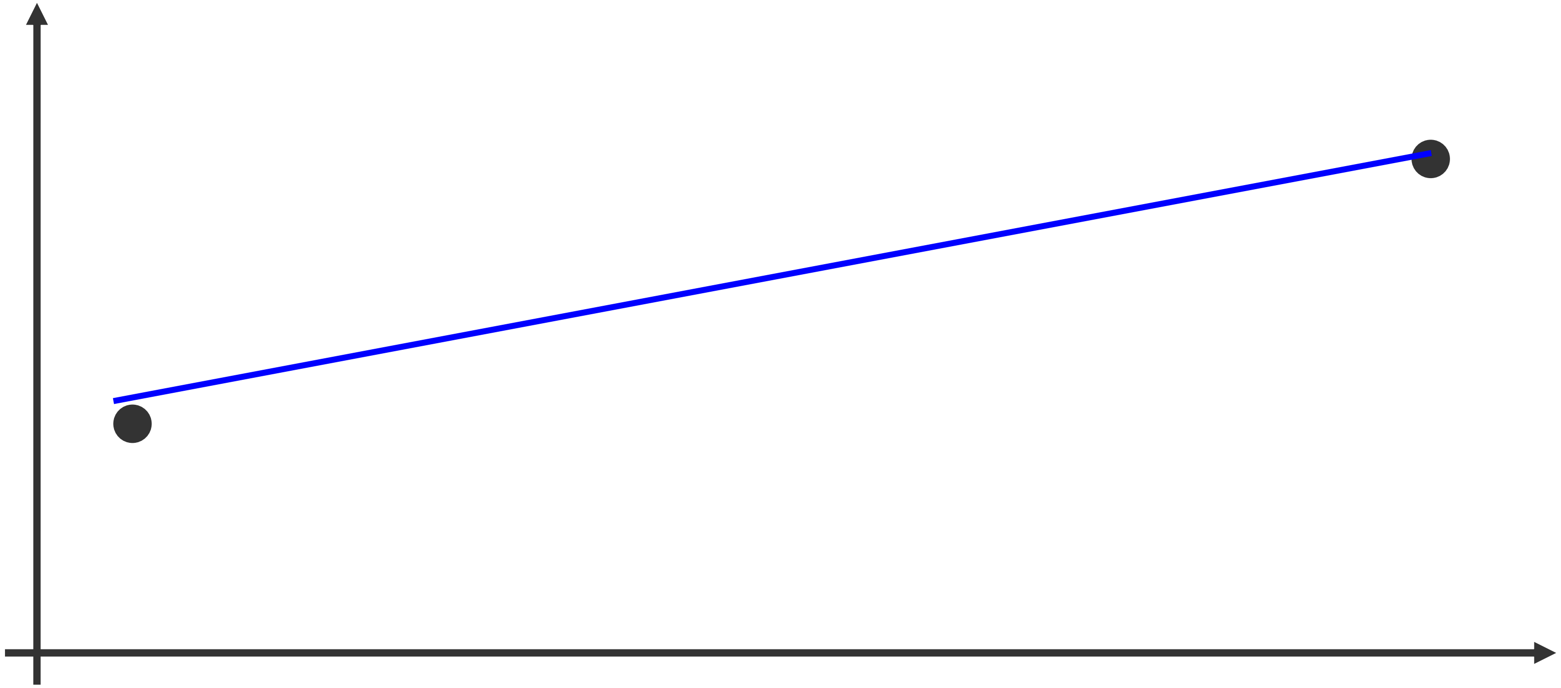
$y^{(i)}$ is the **true class label** of the i-th training example,

$\hat{y}^{(i)}$ is the **predicted class label**

Physics vs. data science

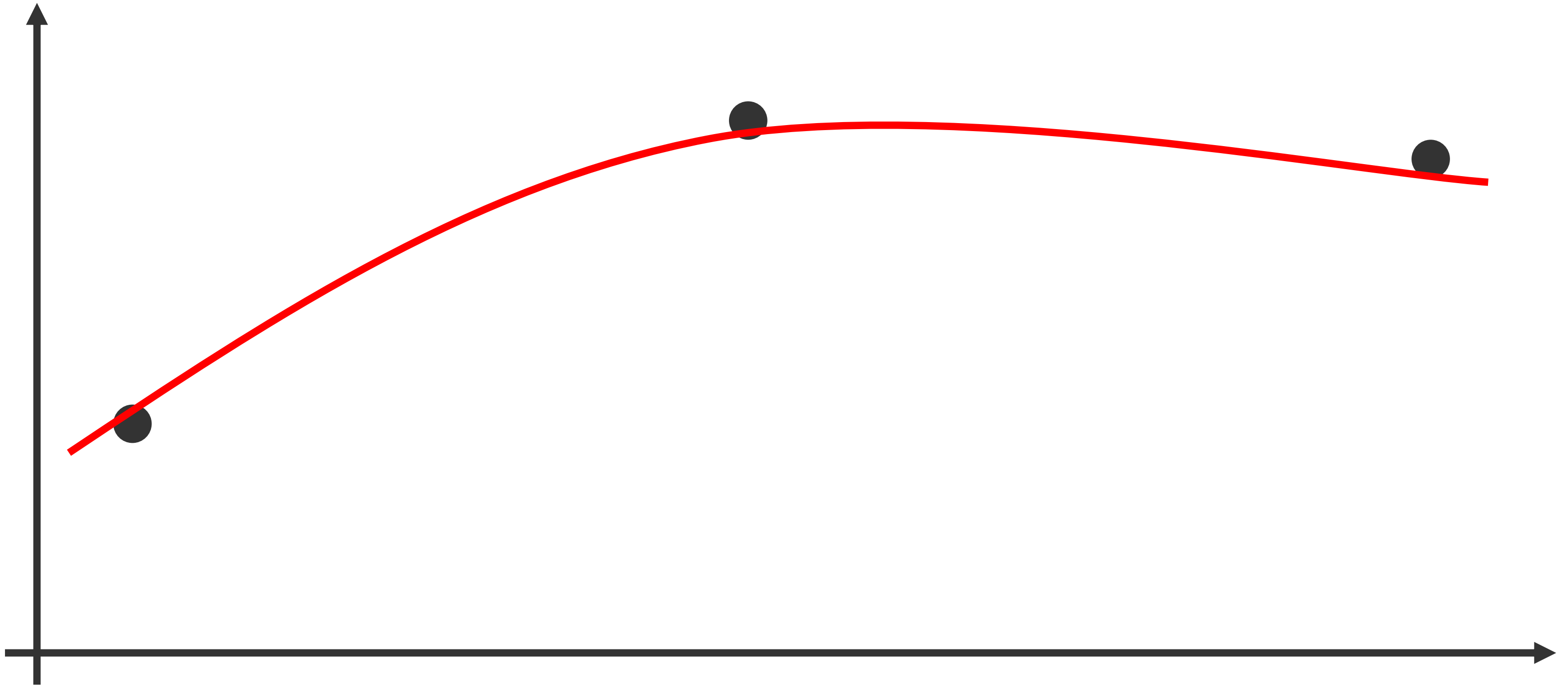


Physics vs. data science



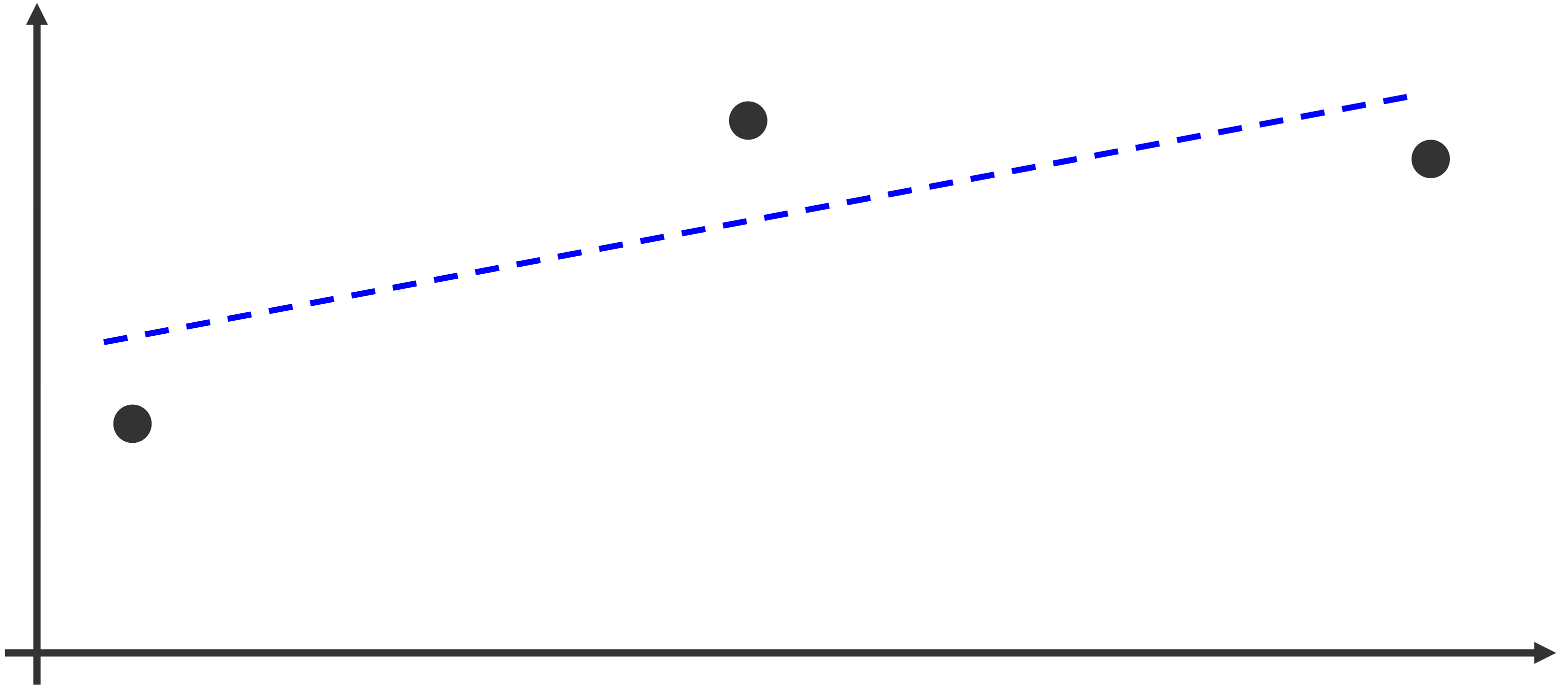
- If we have 2 data points, we “naturally” use linear model
- What should we use if we have three data points? Parabola or linear?
- What if we have one data point?

Physics vs. data science



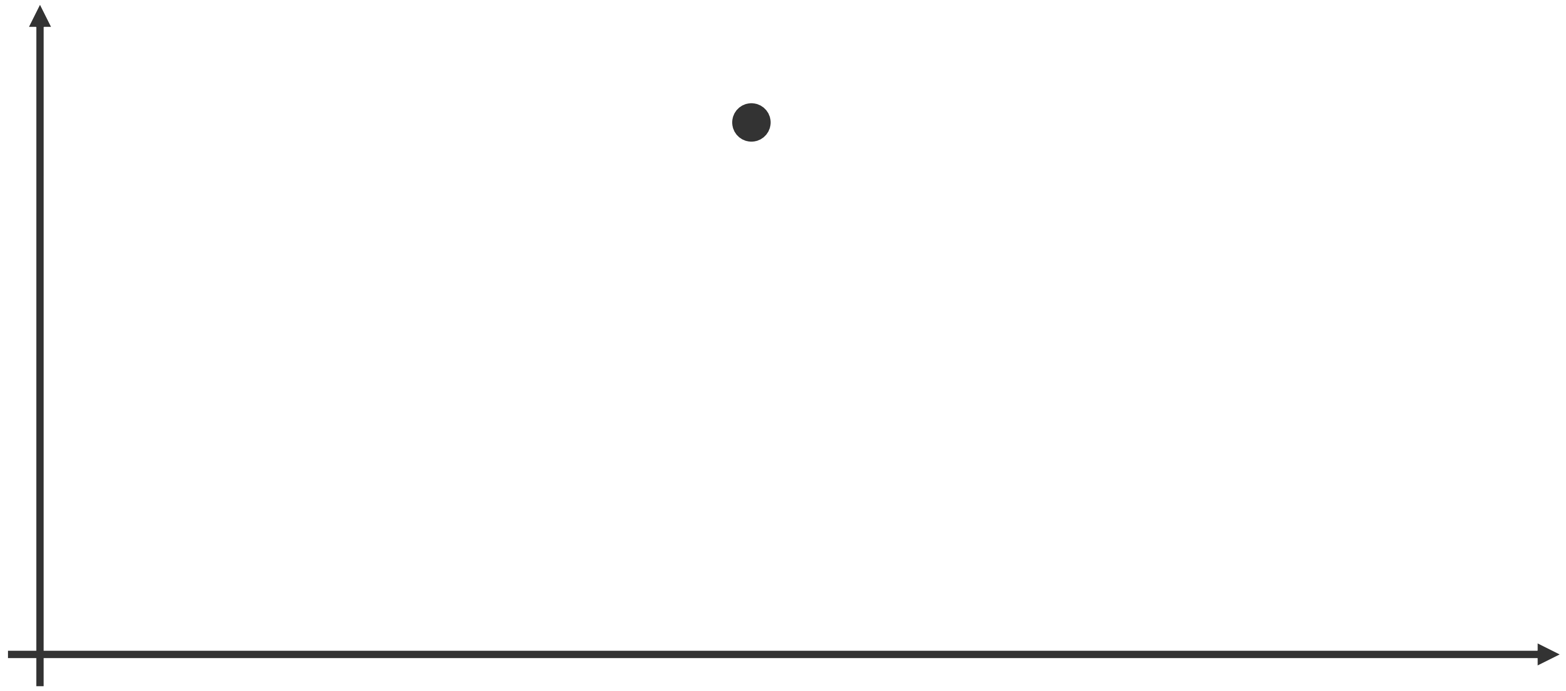
- If we have 2 data points, we “naturally” use linear model
- What should we use if we have three data points? Parabola or linear?
- What if we have one data point?

Physics vs. data science



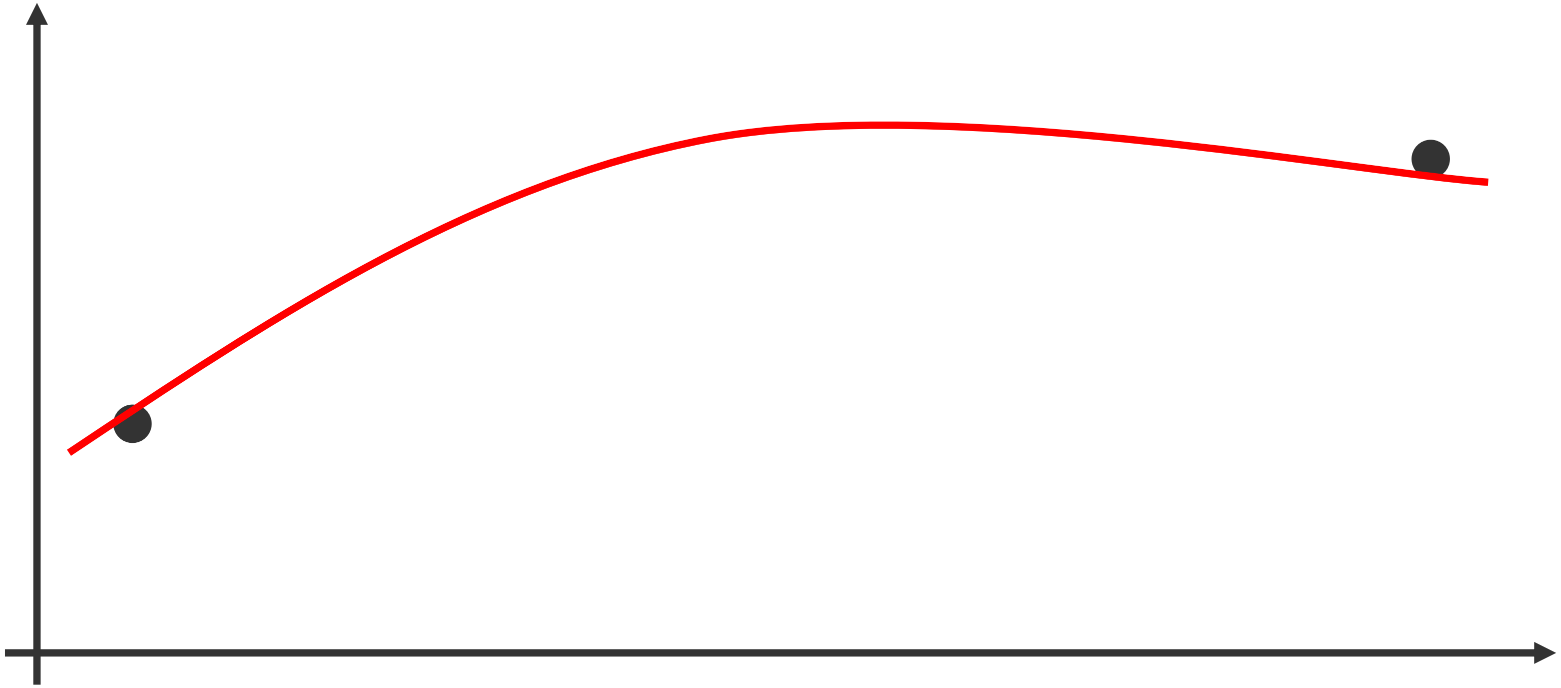
- If we have 2 data points, we “naturally” use linear model
- What should we use if we have three data points? Parabola or linear?
- What if we have one data point?

Physics vs. data science



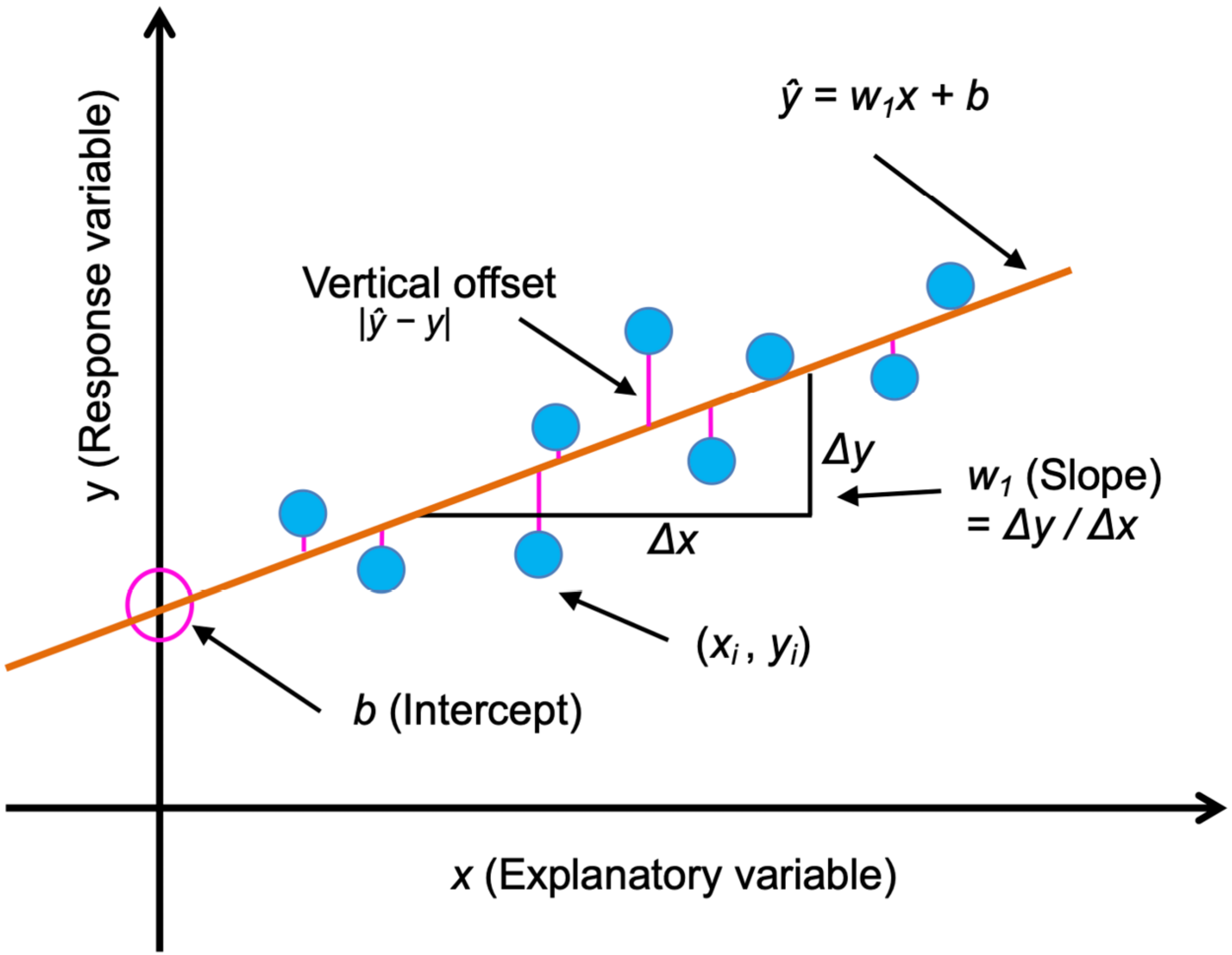
- If we have 2 data points, we “naturally” use linear model
- What should we use if we have three data points? Parabola or linear?
- What if we have one data point?

Physics vs. data science



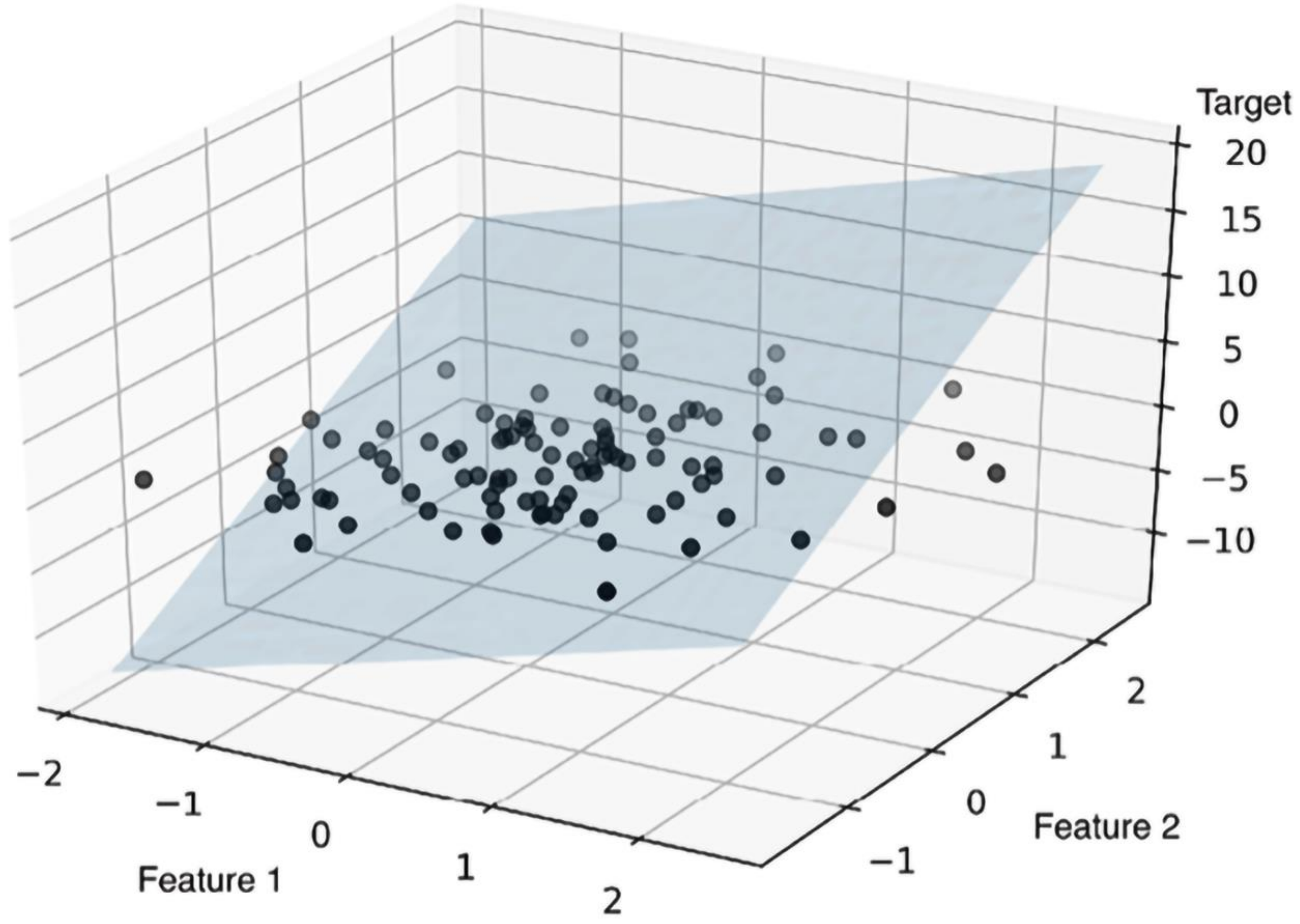
- If we have 2 data points, we “naturally” use linear model
- What should we use if we have three data points? Parabola or linear?
- What if we have one data point?

Linear Regression in 1D



From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

Linear Regression in 2D



From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

RANSAC: Random Sample Consensus

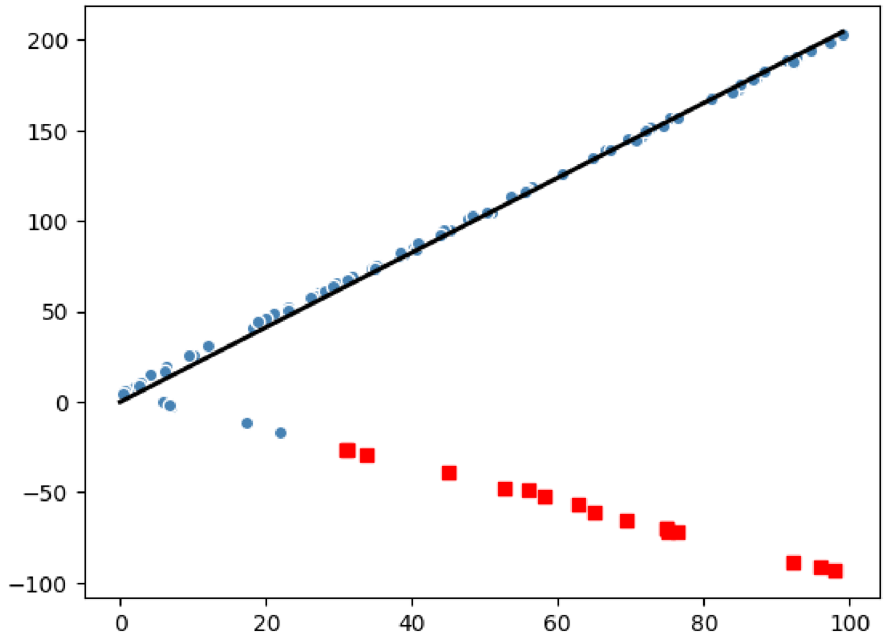
1. Select a random number of examples to be inliers and fit the model.

2. Test all other data points against the fitted model and add those points that fall within a user-given tolerance to the inliers.

3. Refit the model using all inliers.

4. Estimate the error of the fitted model versus the inliers.

5. Terminate the algorithm if the performance meets a certain user-defined threshold or if a fixed number of iterations was reached; go back to *step 1* otherwise.



Regularized linear regression

Linear regression can become nontrivial if **x** in y = lin(**x**) has high D

1. Ridge regression:

$$L(\mathbf{w})_{Ridge} = \sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda||\mathbf{w}||_2^2$$

2. Least absolute shrinkage and selection operator (LASSO):

$$L(\mathbf{w})_{Lasso} = \sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda||\mathbf{w}||_1$$

3. Elastic net

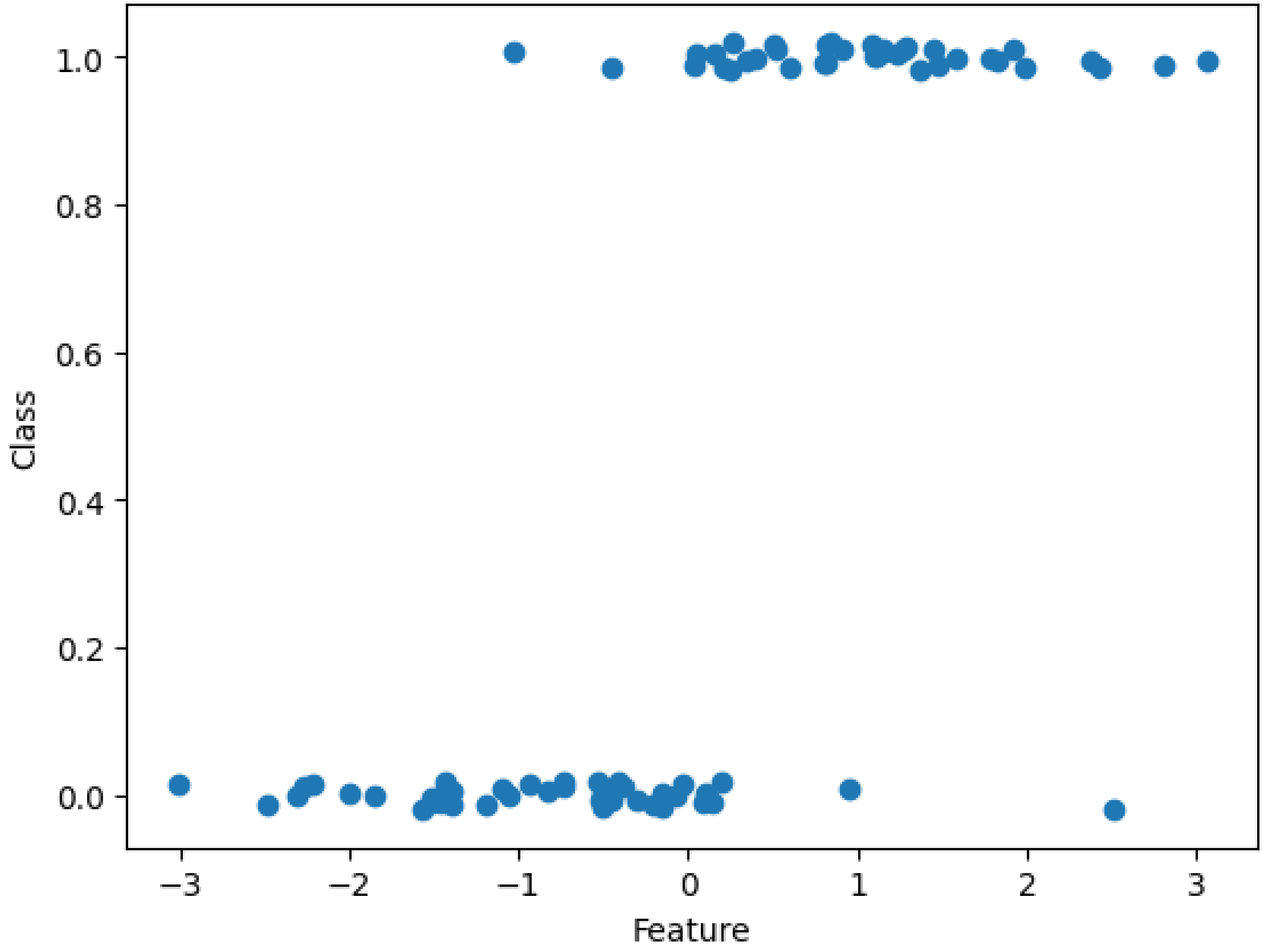
$$L(\mathbf{w})_{Elastic\ Net} = \sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda_2||\mathbf{w}||_2^2 + \lambda_1||\mathbf{w}||_1$$

Practically: need careful consideration of what x is. Dependent on physics, better approach can be DCNNs, causal methods, etc.

Logistic regression

https://www.weknowpets.com.au/blogs/news/are-guinea-pigs-the-right-pet-for-your-family

When do we need logistic regression?



Logistic regression

Probability of event: p

Odds: $\frac{p}{(1-p)}$

Logit: $$\operatorname{logit}(p) = \log \frac{p}{(1-p)}$$

May the odds ever be in your favor!

Logistic model assumes that there is a linear relationship between the weighted inputs and the log-odds

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

Logistic regression

Logistic model:

$$\ln \frac{p}{1-p} = W^T x + b$$

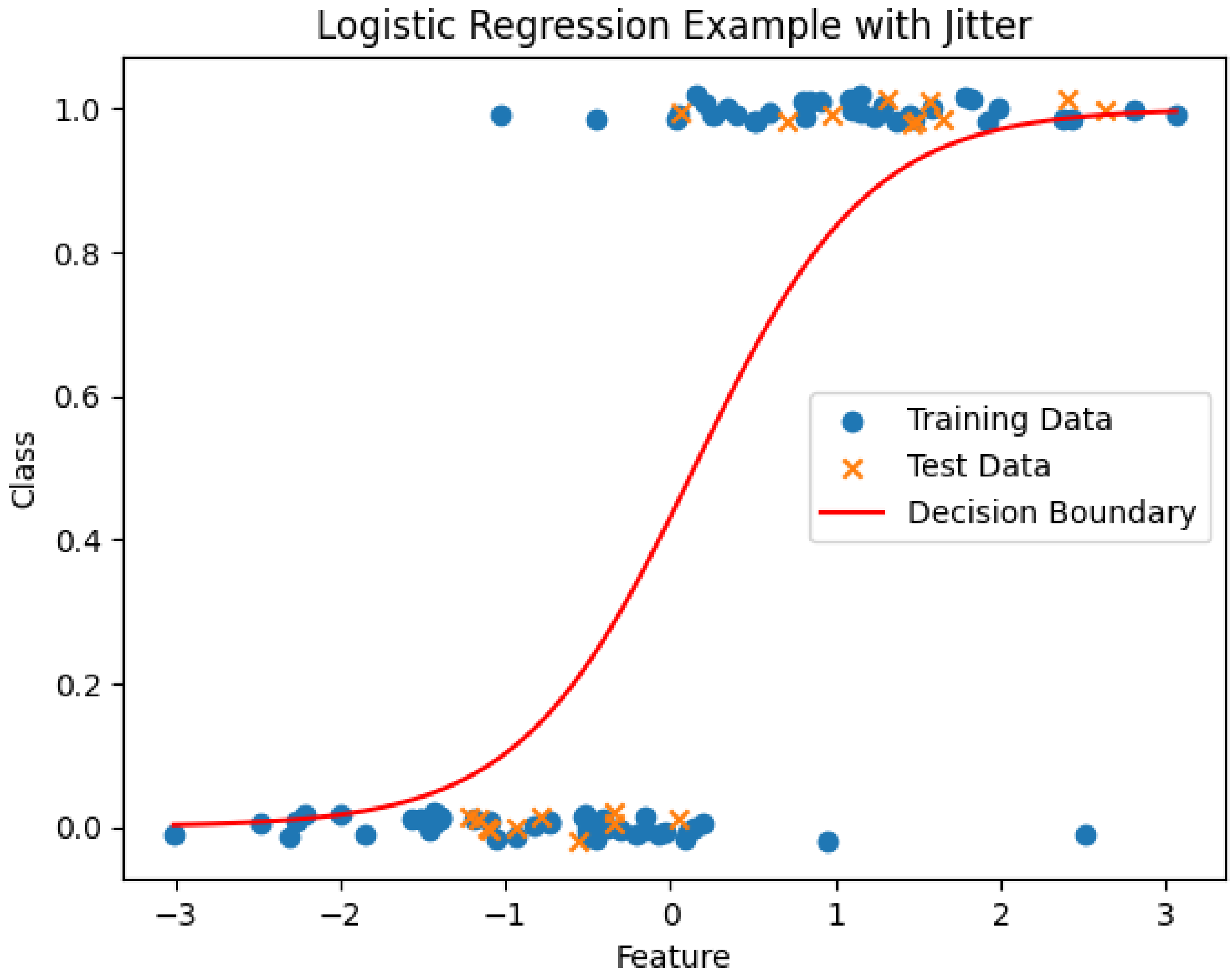
Logistic function:

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

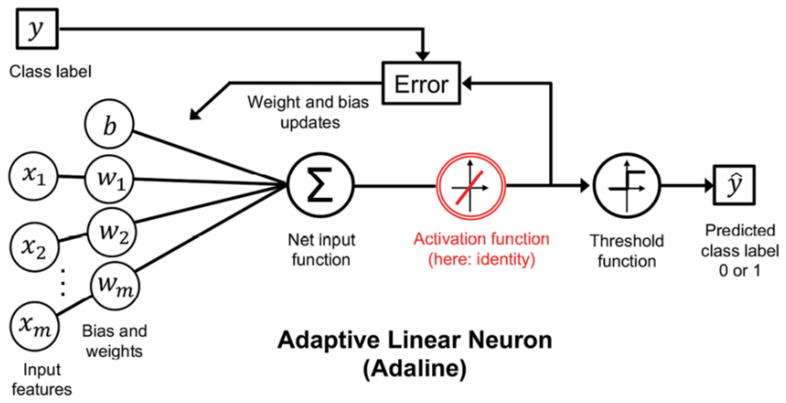
$$z = W^T x + b$$

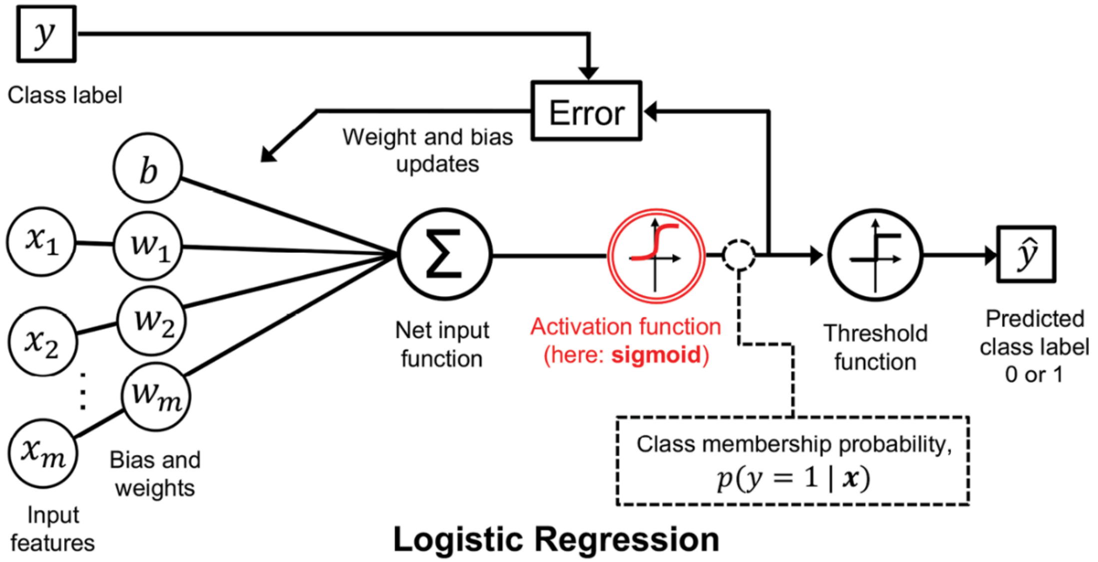
Goal of logistic regression: Predict the "true" proportion of success, p, at any value of the predictor.

Logistic regression prediction



Logistic regression vs. Adaline



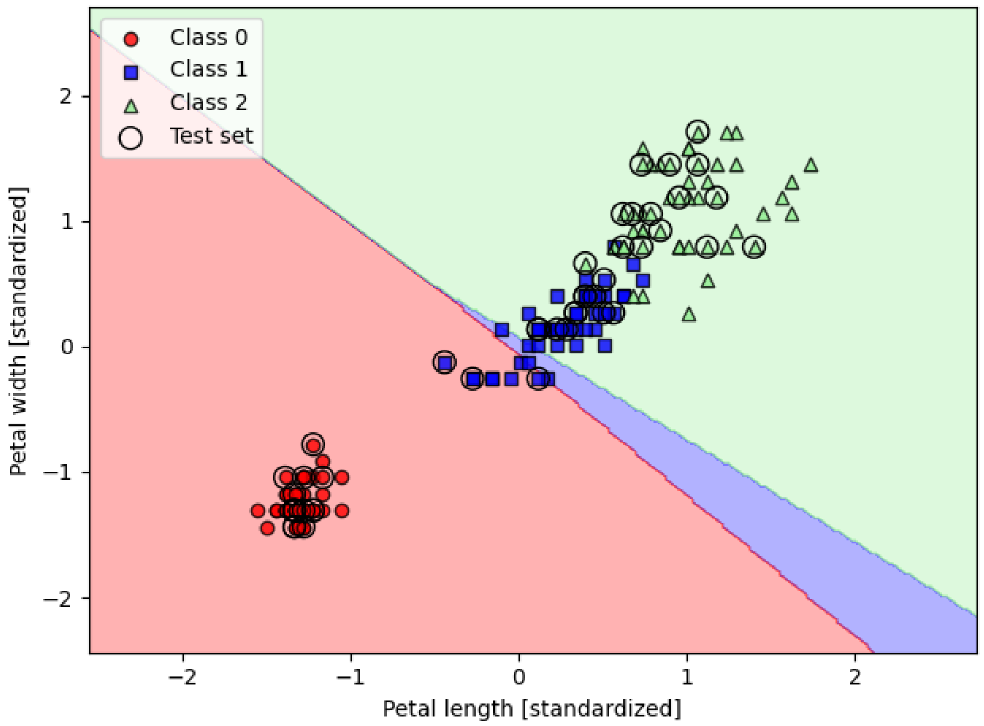


Similar to Adaline, but uses sigmoid as activation function

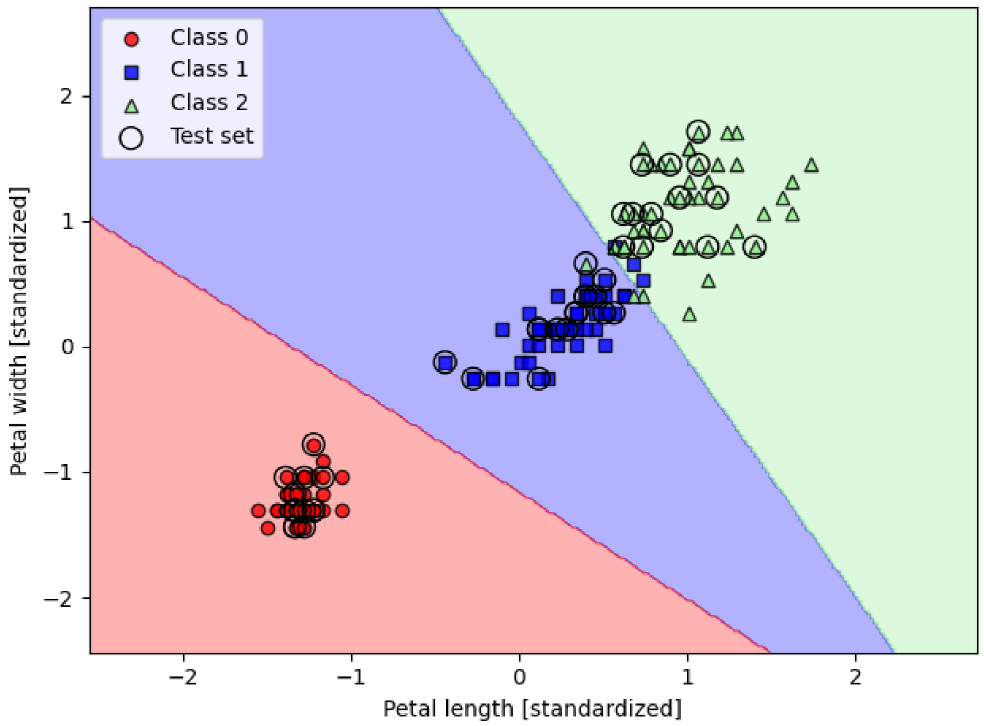
From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

Regularization in logistic regression

C = 0.001



C = 100



1/C

$$L(\boldsymbol{w}, b) = \frac{1}{n}\sum_{i=1}^{n}\left[-y^{(i)}\log\left(\sigma\left(z^{(i)}\right)\right) - \left(1 - y^{(i)}\right)\log\left(1 - \sigma\left(z^{(i)}\right)\right)\right] + \frac{\lambda}{2n}\|\boldsymbol{w}\|^2$$