Advanced Data Analysis

DATA 71200

Class 8: Linear Models

Review from Class 5

- **(Linear) Regression**
  - Continuous predictive model created by estimating a linear relationship between features and response
- **Logistic Regression**
  - Predictive model of the probability of a certain class

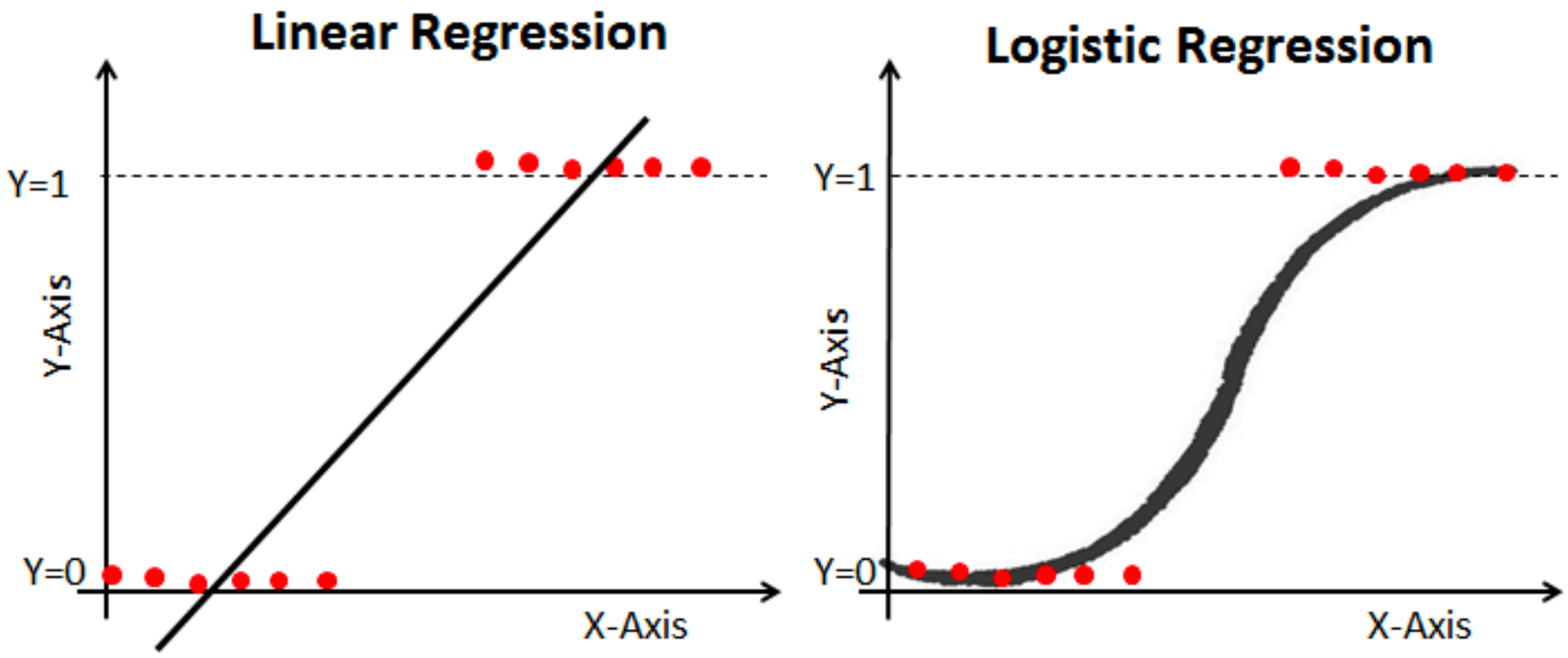


Image: https://medium.com/@hpsuresh12345/logistic-regression-60694a973bee

Linear Models

- **Dimensionality**
  - Line - 1 feature
  - Plane - 2 features
  - Hyperplane - more than 2 features

Linear Regression

Feature weight

Feature value

Predicted Value

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Bias term

Number of features

$\theta_0 = 0$

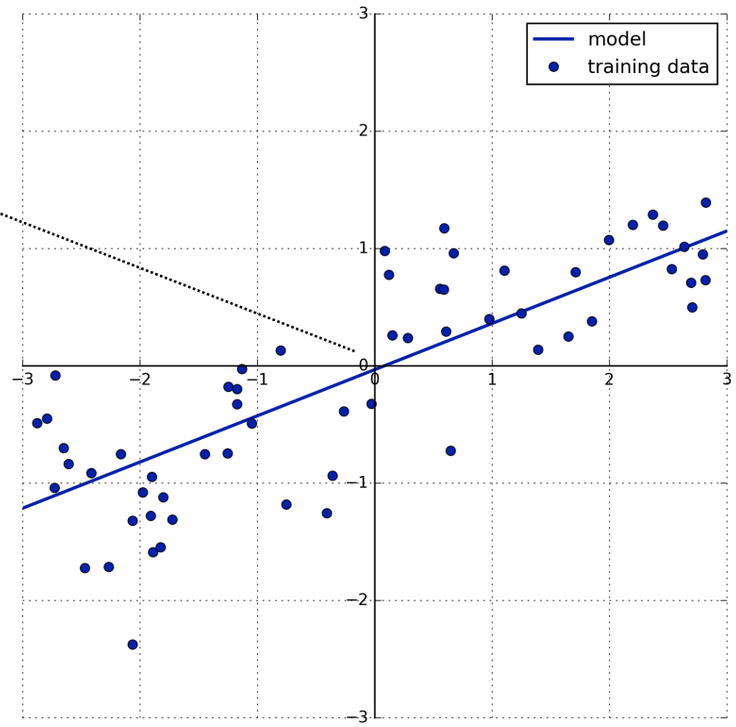
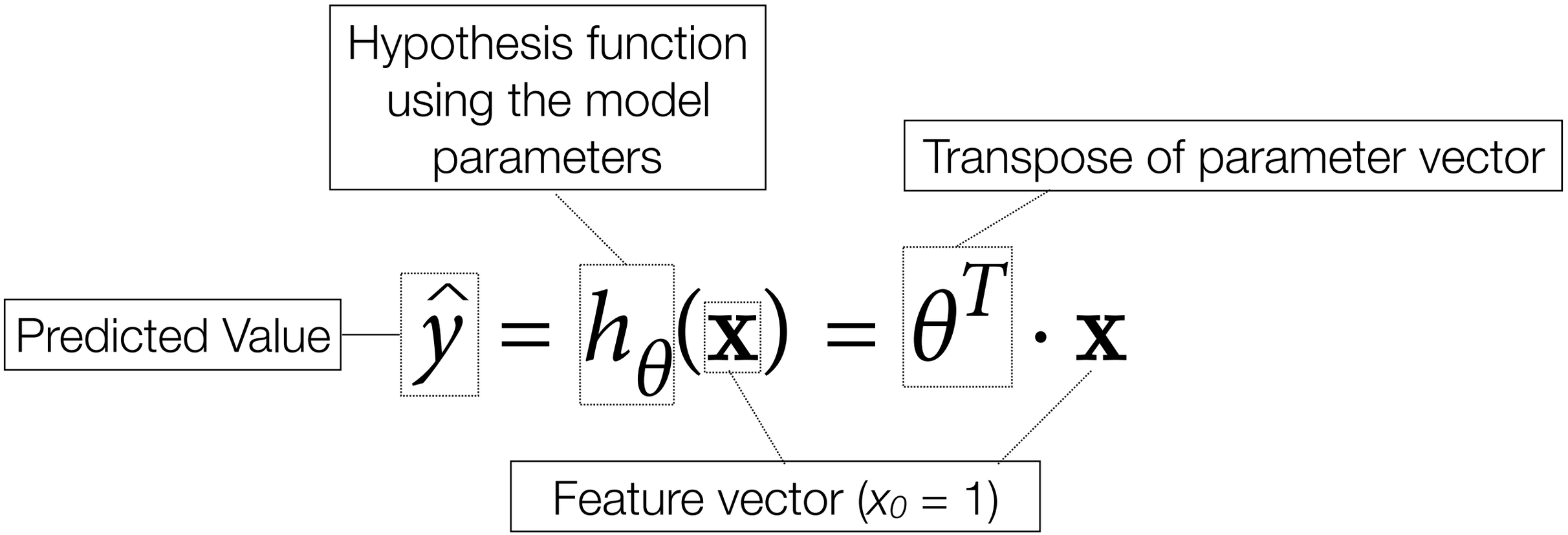


Figure 2-11. Predictions of a linear model on the wave dataset

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

Linear Regression (Vectorized)



Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

The Bias/Variance Tradeoff

An important theoretical result of statistics and Machine Learning is the fact that a model's generalization error can be expressed as the sum of three very different errors:

Bias

This part of the generalization error is due to wrong assumptions, such as assuming that the data is linear when it is actually quadratic. A high-bias model is most likely to underfit the training data.[10]

Variance

This part is due to the model's excessive sensitivity to small variations in the training data. A model with many degrees of freedom (such as a high-degree polynomial model) is likely to have high variance, and thus to overfit the training data.

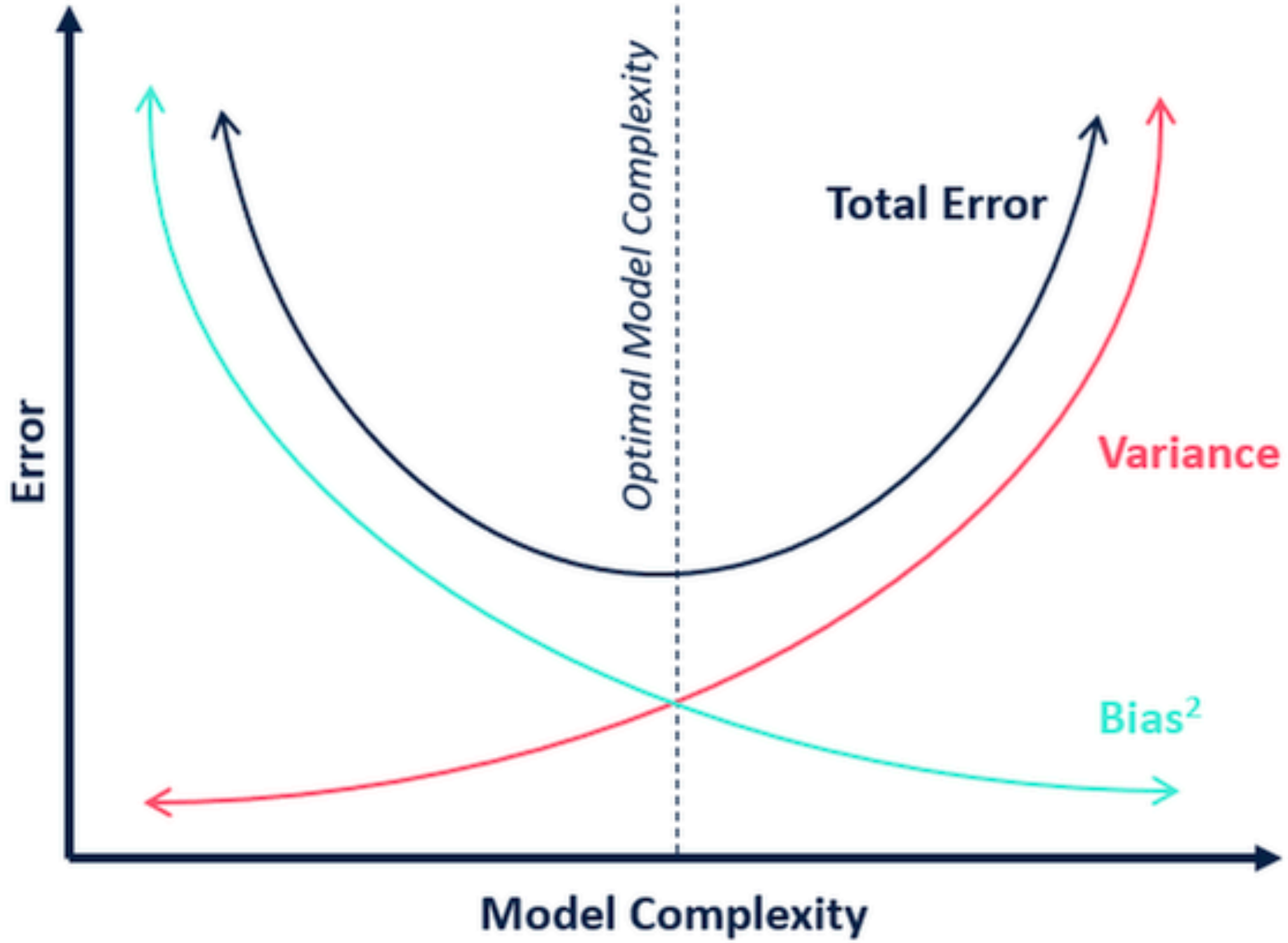
Irreducible error

This part is due to the noisiness of the data itself. The only way to reduce this part of the error is to clean up the data (e.g., fix the data sources, such as broken sensors, or detect and remove outliers).

Increasing a model's complexity will typically increase its variance and reduce its bias. Conversely, reducing a model's complexity increases its bias and reduces its variance. This is why it is called a tradeoff.

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

Bias/Variance Tradeoff



https://ai-pool.com/a/s/bias-variance-tradeoff-in-machine-learning

MSE Cost Function

Equation 4-3. MSE cost function for a Linear Regression model

$$\mathrm{MSE}(\mathbf{X}, h_\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(\theta^T \cdot \mathbf{x}^{(i)} - y^{(i)}\right)^2$$

X

- **X** is a matrix containing all the feature values (excluding labels) of all instances in the dataset. There is one row per instance and the i^{th} row is equal to the transpose of $\mathbf{x}^{(i)}$, noted $(\mathbf{x}^{(i)})^T$.[6]
  - For example, if the first district is as just described, then the matrix **X** looks like this:

$$\mathbf{X} = \begin{pmatrix} \left(\mathbf{x}^{(1)}\right)^T \\ \left(\mathbf{x}^{(2)}\right)^T \\ \vdots \\ \left(\mathbf{x}^{(1999)}\right)^T \\ \left(\mathbf{x}^{(2000)}\right)^T \end{pmatrix} = \begin{pmatrix} -118.29 & 33.91 & 1,416 & 38,372 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

h_θ

- h_θ is the hypothesis function, using the model parameters θ.

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

MSE Cost Function

$$\mathrm{MSE}(\mathbf{X}, h_\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(\theta^T \cdot \mathbf{x}^{(i)} - y^{(i)}\right)^2$$

- m: Number of data instances
- θ^T: Transpose of parameter vector
- $\mathbf{x}^{(i)}$: i^{th} instance in vector of feature
- $y^{(i)}$: i^{th} instance in vector of labels

Also referred to as Ordinary Least Squares (OLS)

Jupyter Notebook 02-supervised-learning .ipynb [25-29]

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

Ridge Regression Cost Function

Cost function

Tunable parameter (alpha)

Number of features (dimensions)

$$J(\theta) = \text{MSE}(\theta) + \alpha \frac{1}{2} \sum_{i=1}^{n} \theta_i^2$$

$$\text{MSE}(\mathbf{X}, h_\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(\theta^T \cdot \mathbf{x}^{(i)} - y^{(i)} \right)^2$$

Jupyter Notebook
02-supervised-learning
.ipynb [30-32]

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

Ridge Regression

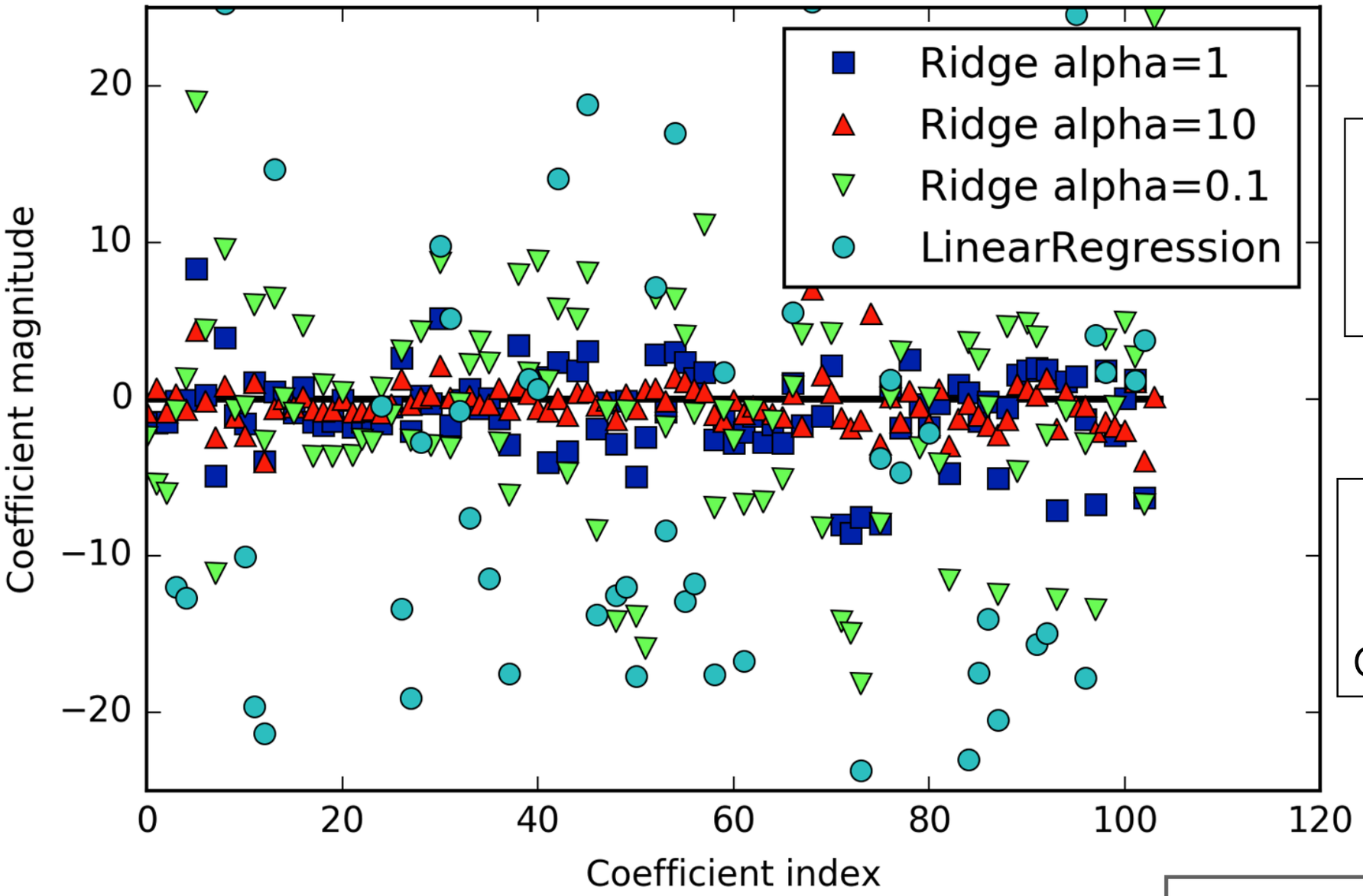


Figure 2-12. Comparing coefficient magnitudes for ridge regression with different values of alpha and linear regression

Trade off between model simplicity and model performance

Increasing alpha moves more coefficients towards 0

Jupyter Notebook 02-supervised-learning .ipynb [33]

Guido, Sarah and Andreas C. Muller. (2016). Introduction to Machine Learning with Python, O'Reilly Media, Inc.

Ridge Regression

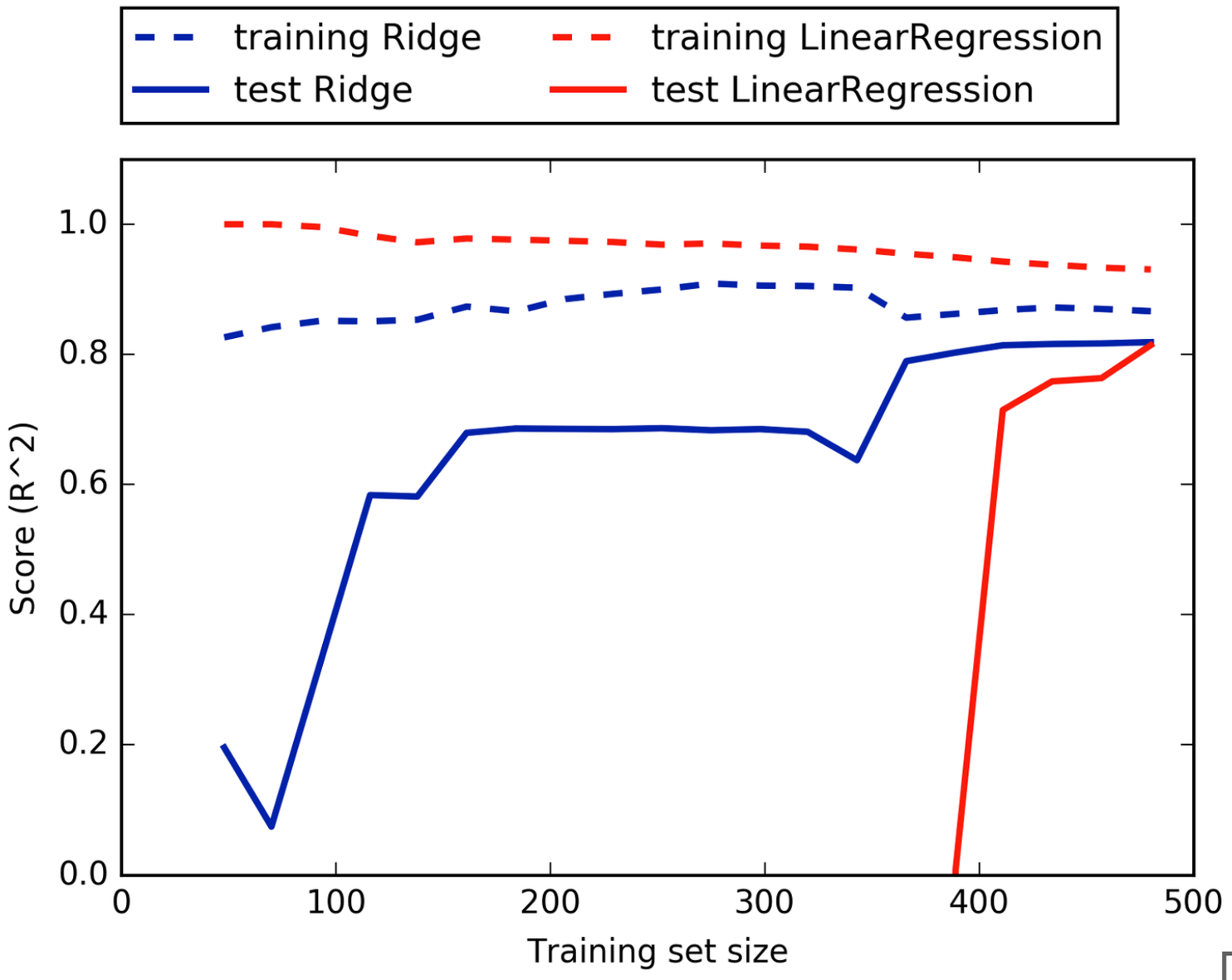


Figure 2-13. Learning curves for ridge regression and linear regression on the Boston Housing dataset

Ridge performs worse than OLS on training set but better on testing set

OLS improves performance on testing set with more data (generalizes better so performance on training data declines)

Jupyter Notebook 02-supervised-learning .ipynb [34]

Guido, Sarah and Andreas C. Muller. (2016). Introduction to Machine Learning with Python, O'Reilly Media, Inc.

Lasso Regression Cost Function

Tunable parameter (alpha)

Number of features (dimensions)

$$J(\theta) = \text{MSE}(\theta) + \alpha \sum_{i=1}^{n} |\theta_i|$$

$$\text{MSE}(\mathbf{X}, h_\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(\theta^T \cdot \mathbf{x}^{(i)} - y^{(i)}\right)^2$$

Recall the Ridge Equation

$$J(\theta) = \text{MSE}(\theta) + \alpha \frac{1}{2} \sum_{i=1}^{n} \theta_i^2$$

Jupyter Notebook 02-supervised-learning .ipynb [35-37]

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

Lasso versus Ridge Regression

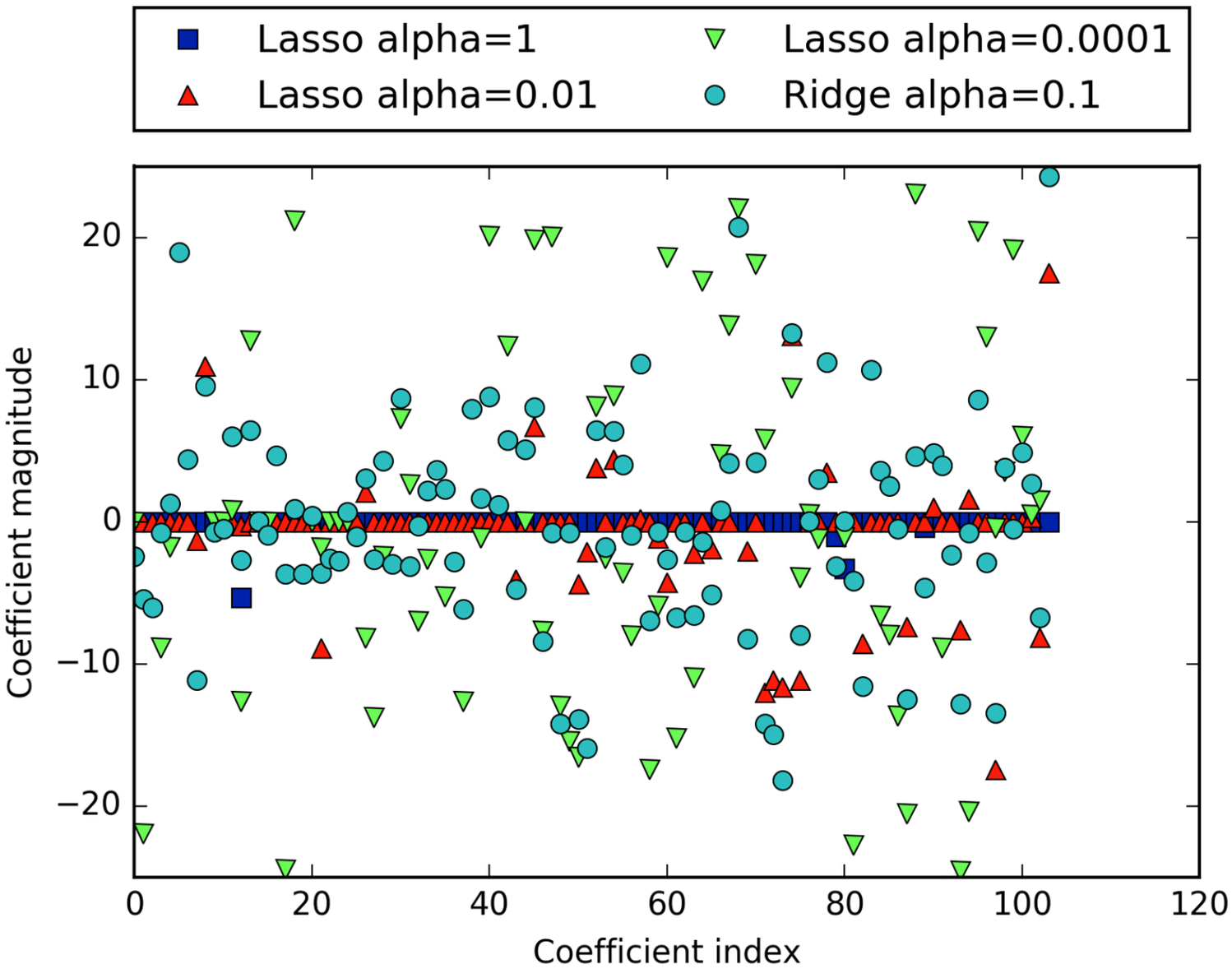


Figure 2-14. Comparing coefficient magnitudes for lasso regression with different values of alpha and ridge regression

When alpha is too high, too many features are zeroed out

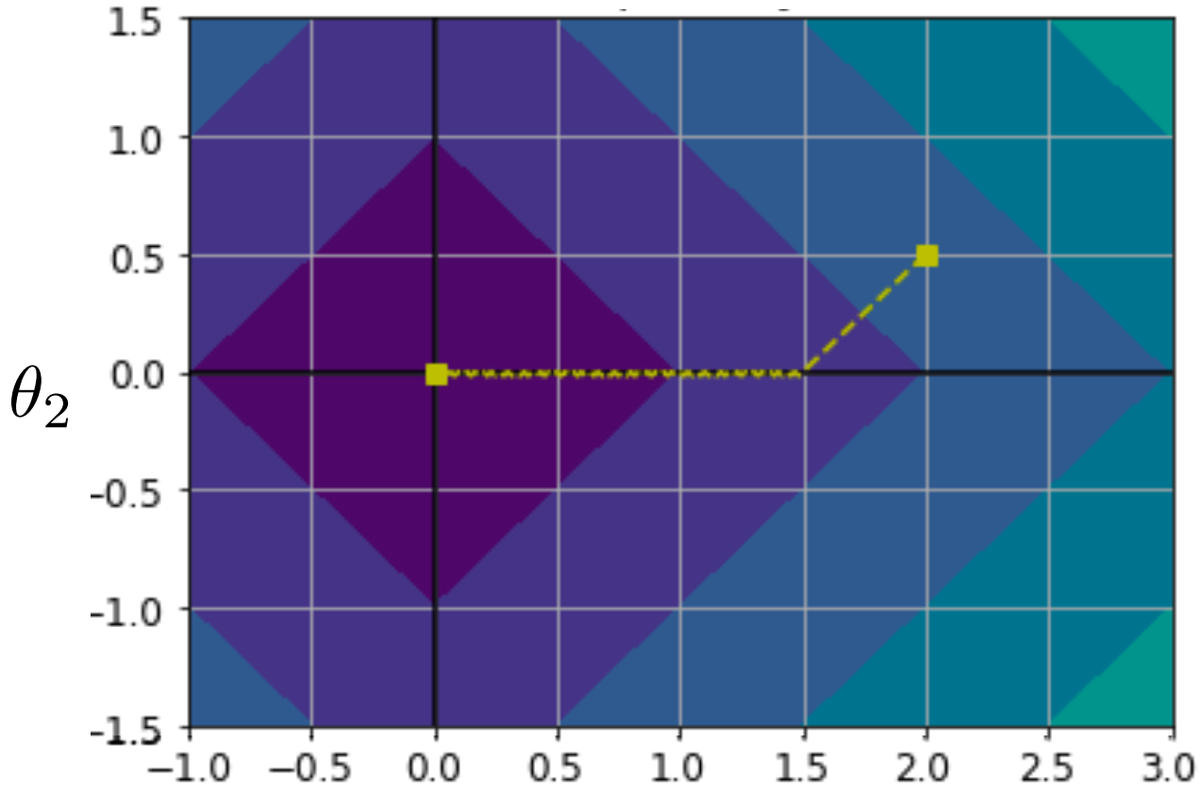
When alpha is too low, the model is effectively unregularized

Lasso alpha=0.01 is similar to Ridge alpha=0.1 but some coefficients are zeroed out

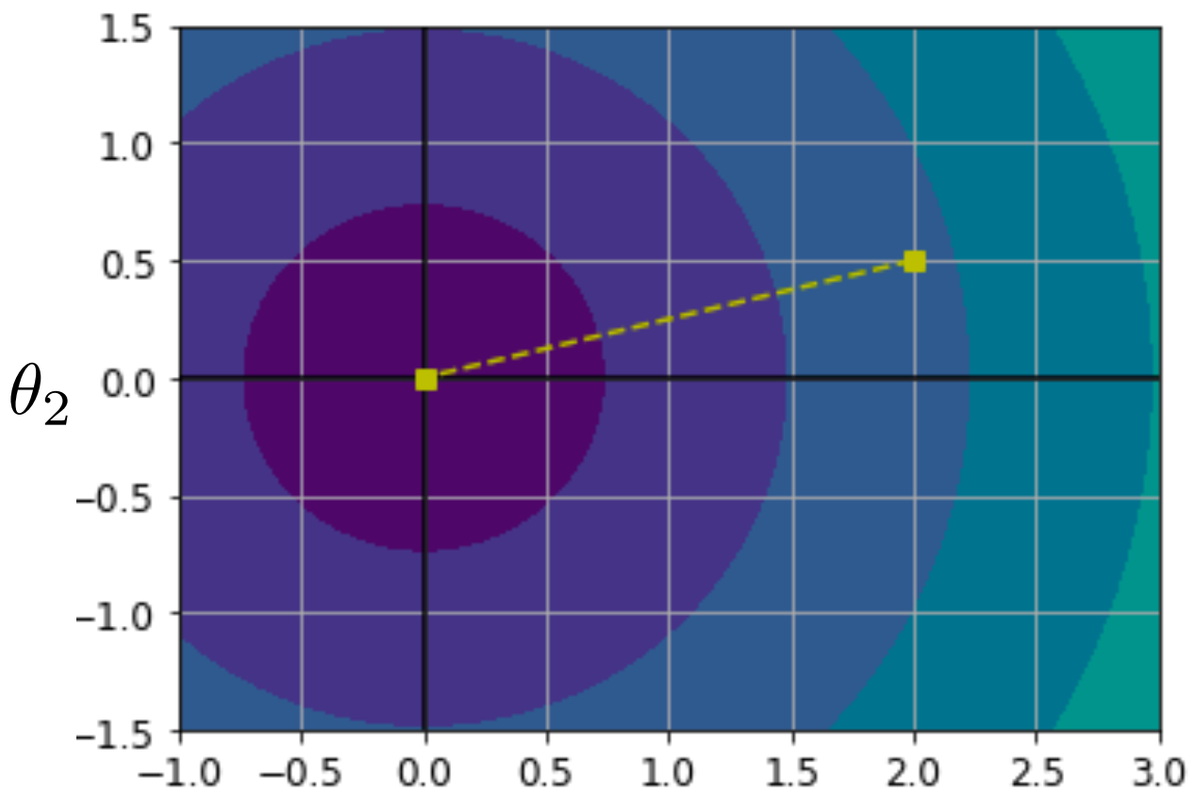
Jupyter Notebook 02-supervised-learning .ipynb [38]

Lasso versus Ridge Regression

ℓ_1 penalty (Lasso)



ℓ_2 penalty (Ridge)



Observe that in Lasso θ_2 remains 0 until it reaches a minimum value (0.5) while in Ridge the value of θ_2 can be a value between 0 and 0.5

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

Logistic Regression (Vectorized)

Linear Regression

$$\hat{y} = h_\theta(\mathbf{x}) = \theta^T \cdot \mathbf{x}$$

Hypothesis function using the model parameters

Logistic function

Transpose of parameter vector

Estimated probability

$$\hat{p} = h_\theta(\mathbf{x}) = \sigma\left(\theta^T \cdot \mathbf{x}\right)$$

Feature vector ($x_0 = 1$)

Logistic Function

$$\sigma(t) = \frac{1}{1 + \exp(-t)}$$

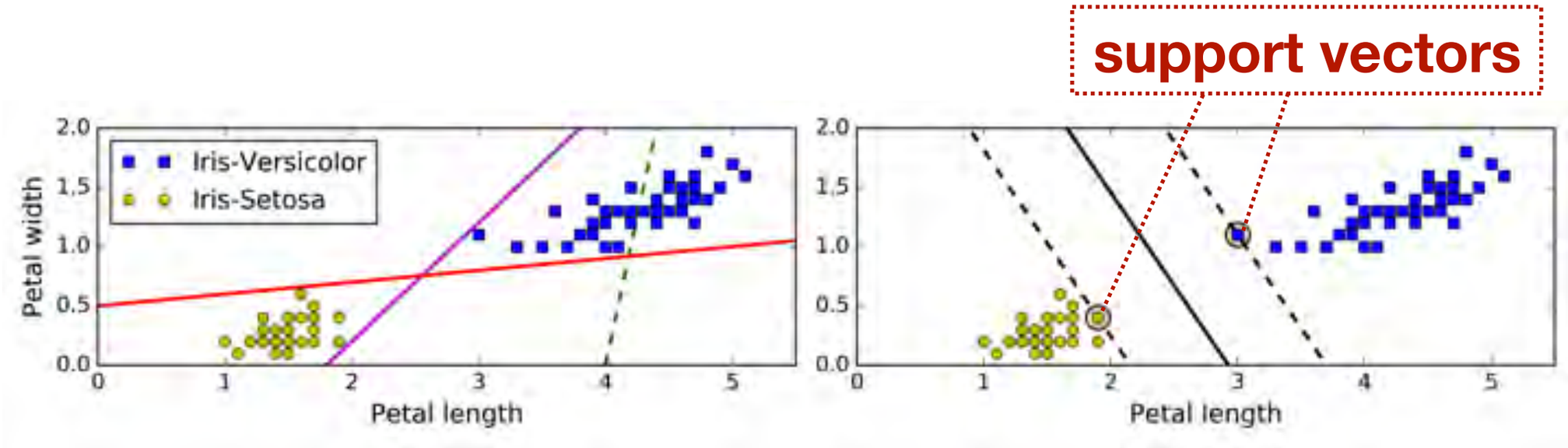
Predicted Class (binary)

$$\hat{y} = \begin{cases} 0 \text{ if } \hat{p} < 0.5, \\ 1 \text{ if } \hat{p} \geq 0.5. \end{cases}$$

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

Linear Support Vector Classifier

- **Find the linear classifier with the best separation (margin) between the two classes**
  - Operationally the data points used to make this calculation are the support vectors



Three possible linear classifiers

Linear classifier with the widest margin

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

Logistic Reg versus Linear SVC

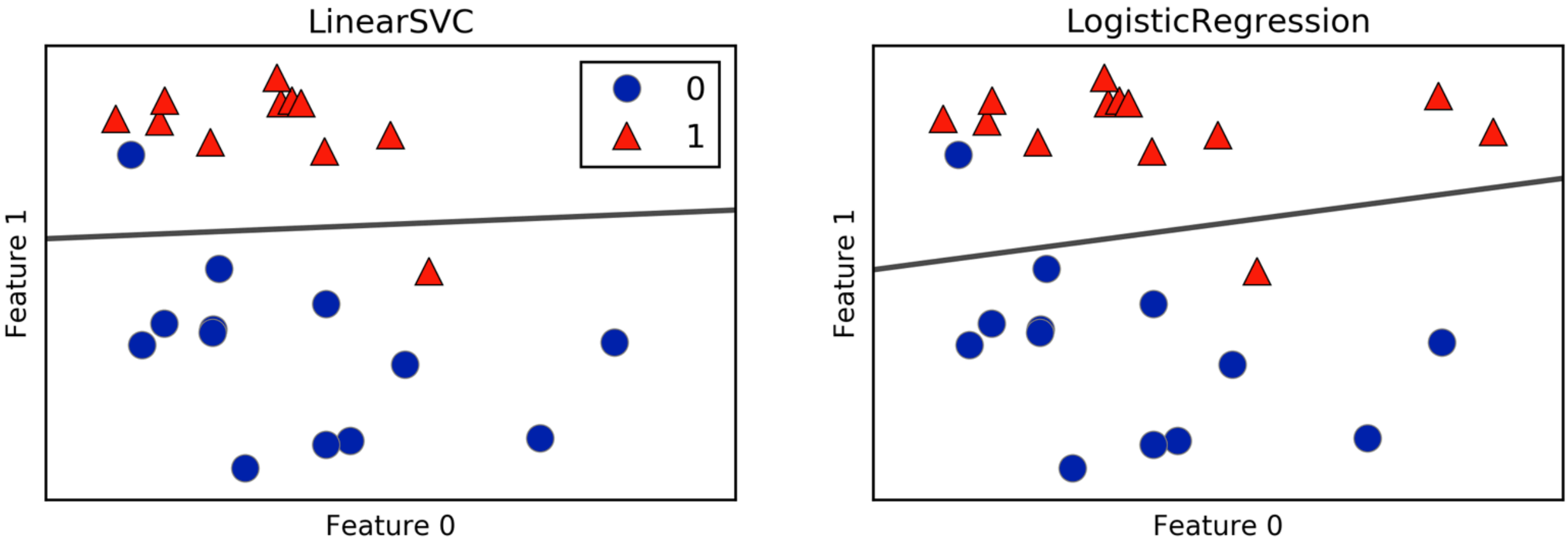


Figure 2-15. Decision boundaries of a linear SVM and logistic regression on the forge dataset with the default parameters

Jupyter Notebook
02-supervised-learning
.ipynb [39]

Coefficient Magnitude

- **C is the inverse of regularization strength**

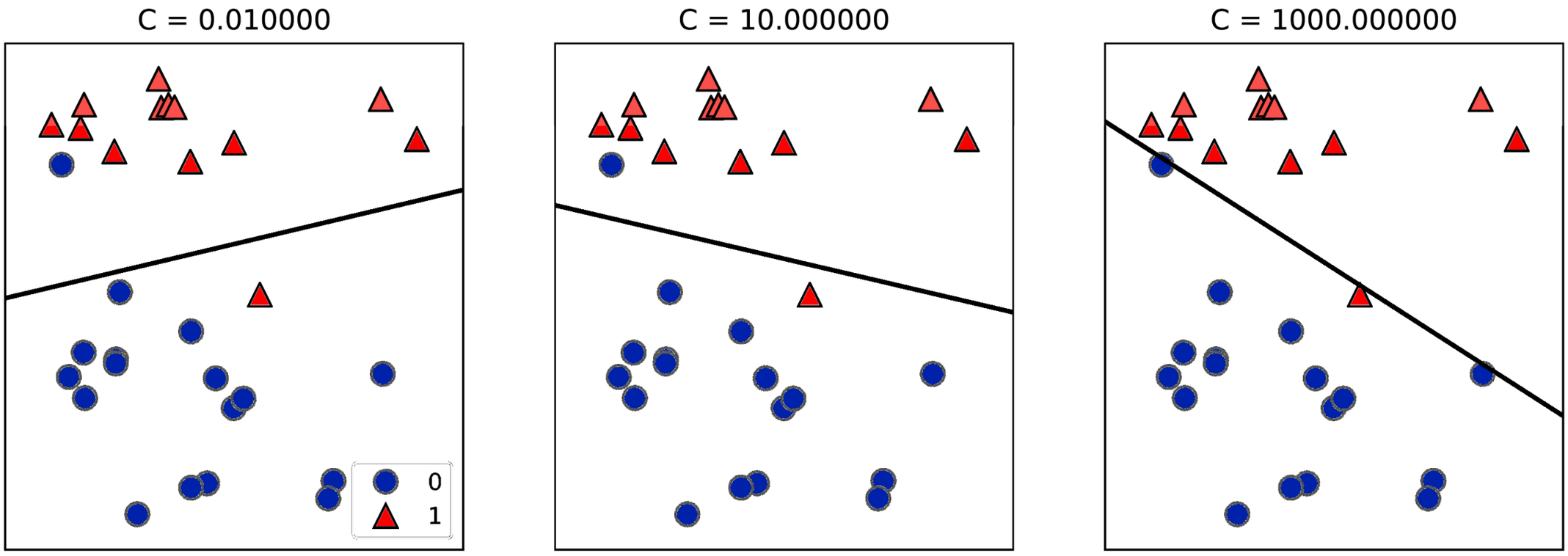


Figure 2-16. Decision boundaries of a linear SVM on the forge dataset for different values of C

Low values of C - algorithm adjusts to the majority of the data points

High values of C - algorithm attempts to correctly classify as many individual data points as possible

Jupyter Notebook 02-supervised-learning .ipynb [40]

Coefficient Magnitude

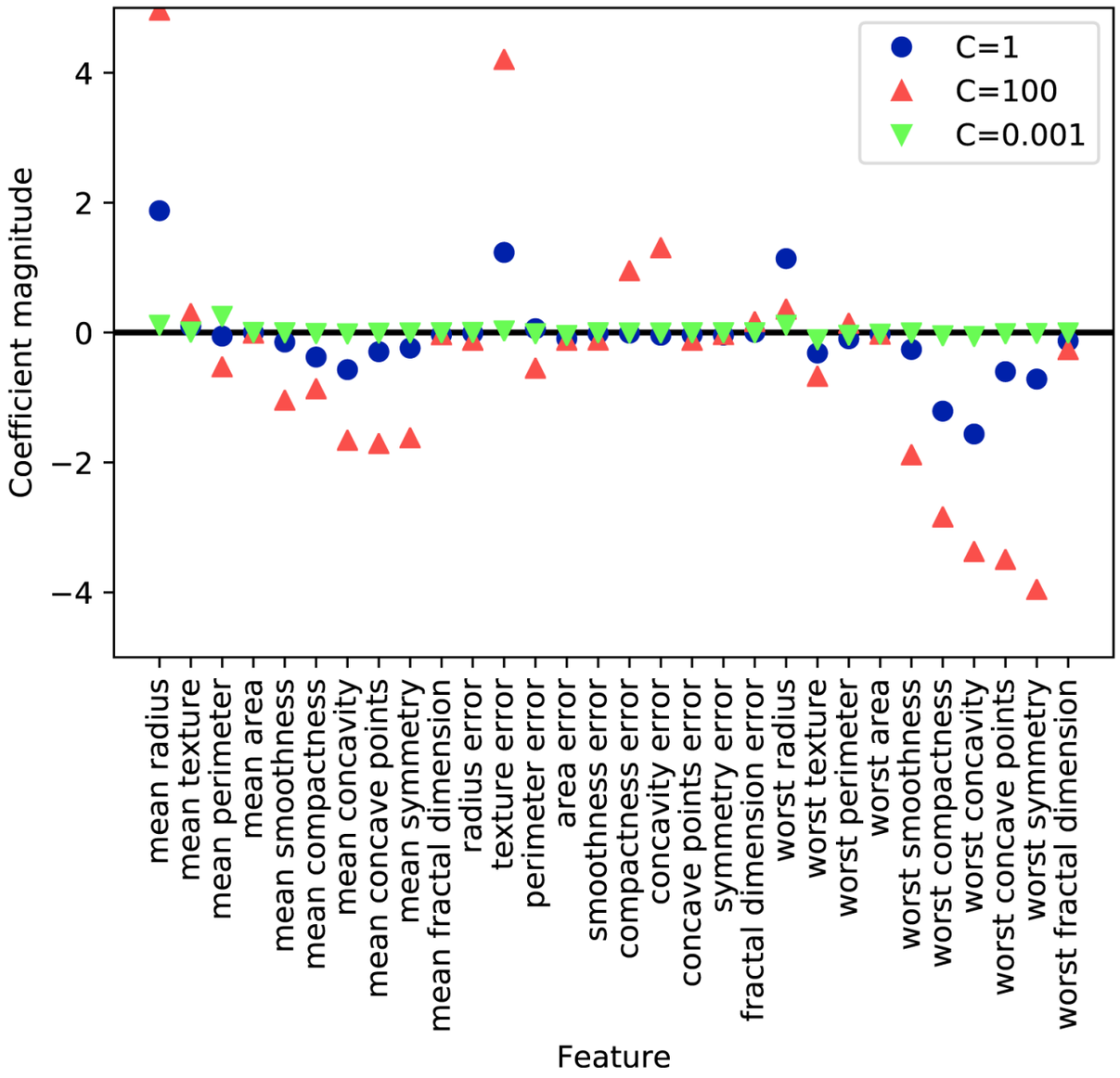


Figure 2-17. Coefficients learned by logistic regression on the Breast Cancer dataset for different values of C

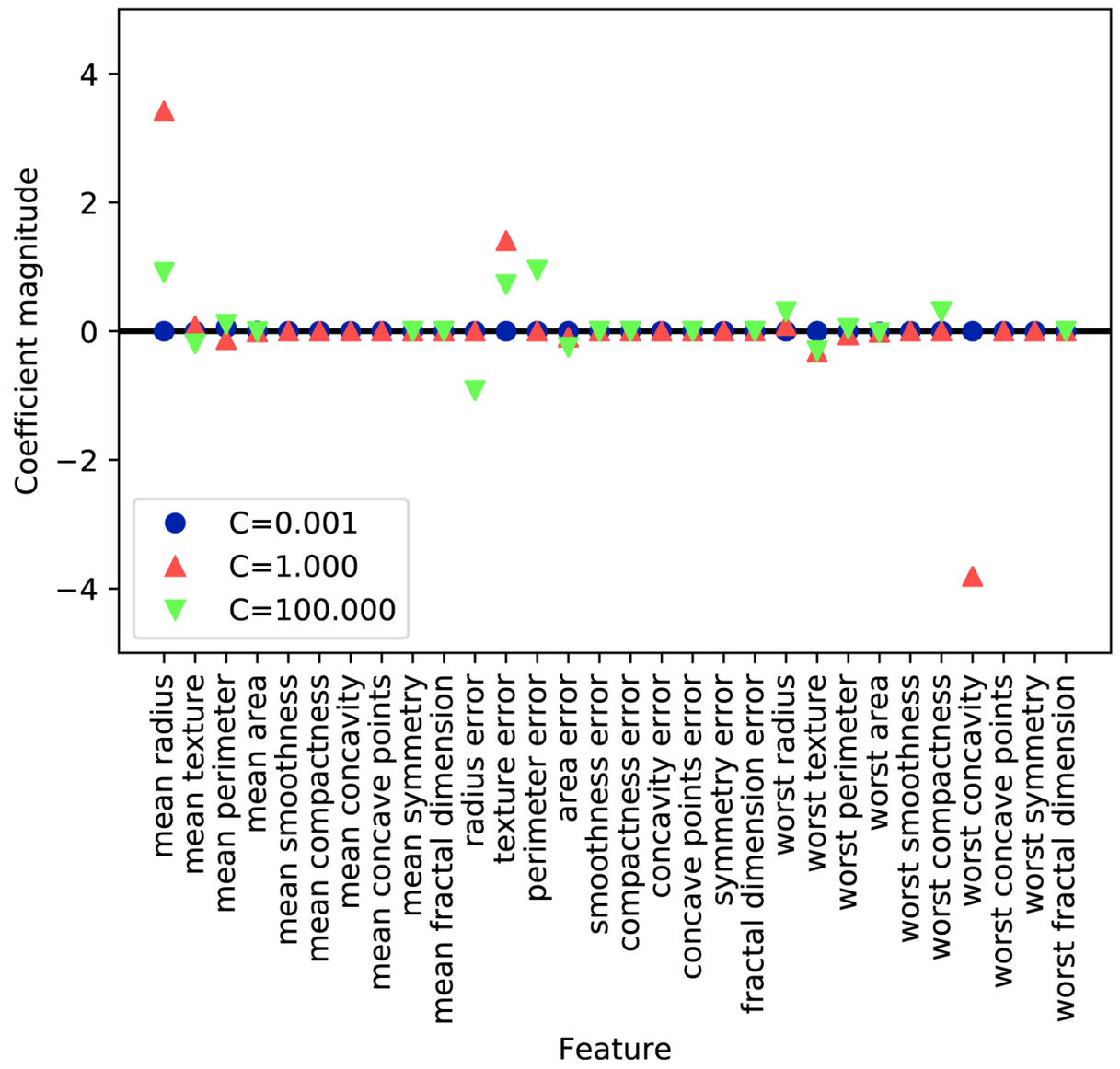


Figure 2-18. Coefficients learned by logistic regression with L1 penalty on the Breast Cancer dataset for different values of C

Jupyter Notebook 02-supervised-learning.ipynb [41-45]

One versus Rest Classification

- **Binary model is learned for each class**
- **All models run on each test point - the model with the highest score "wins"**

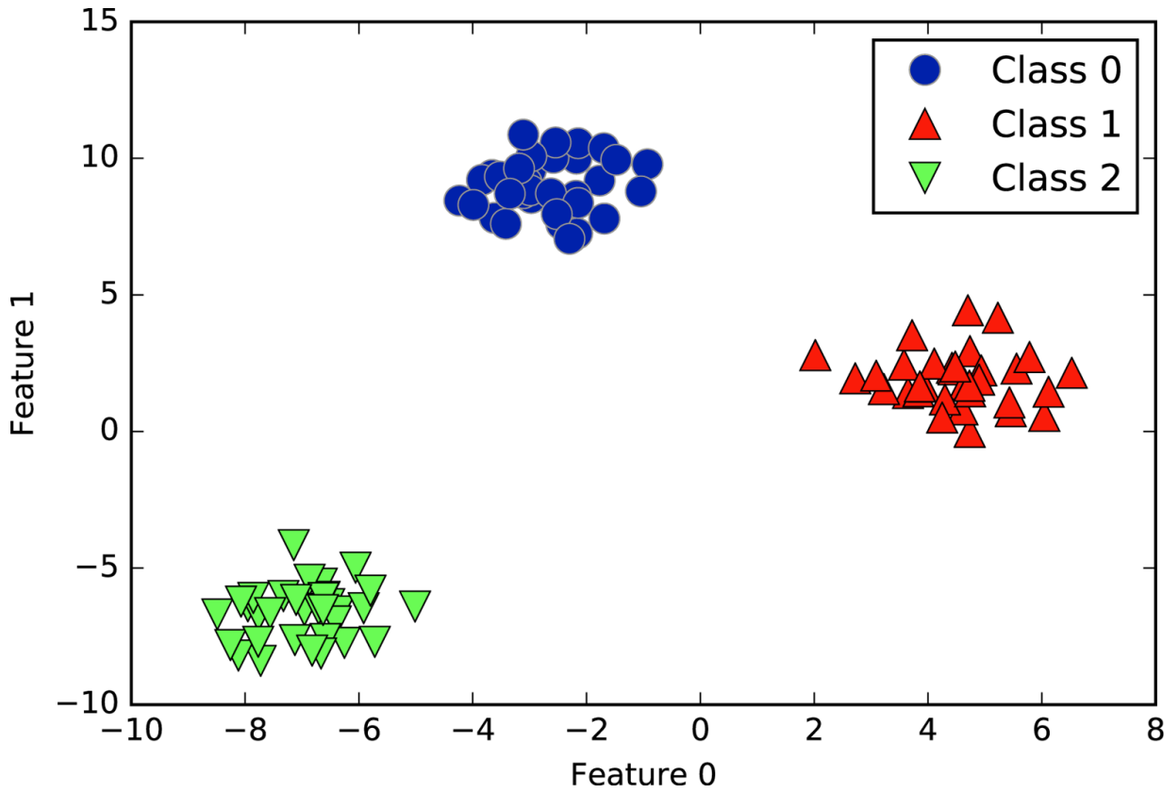


Figure 2-19. Two-dimensional toy dataset containing three classes

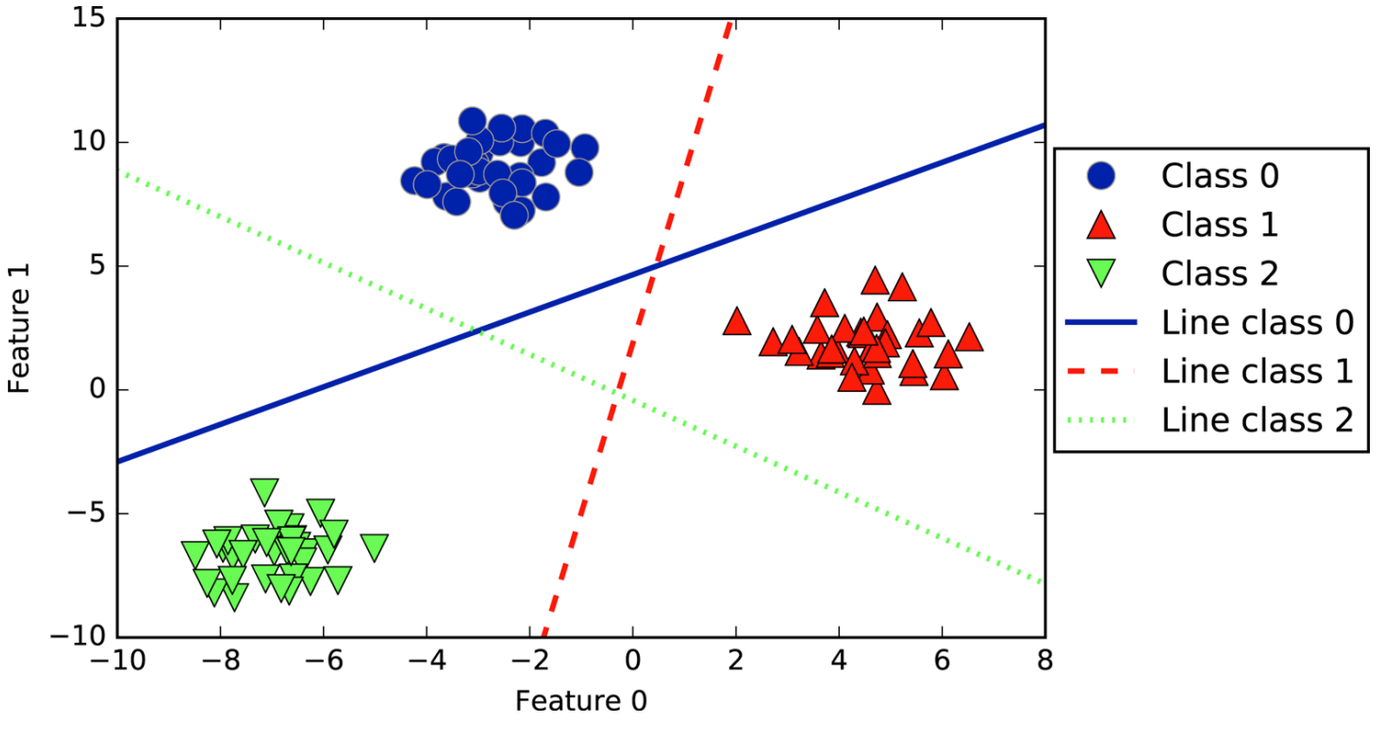


Figure 2-20. Decision boundaries learned by the three one-vs.-rest classifiers

Jupyter Notebook 02-supervised-learning .ipynb [46-48]

One versus Rest Classification

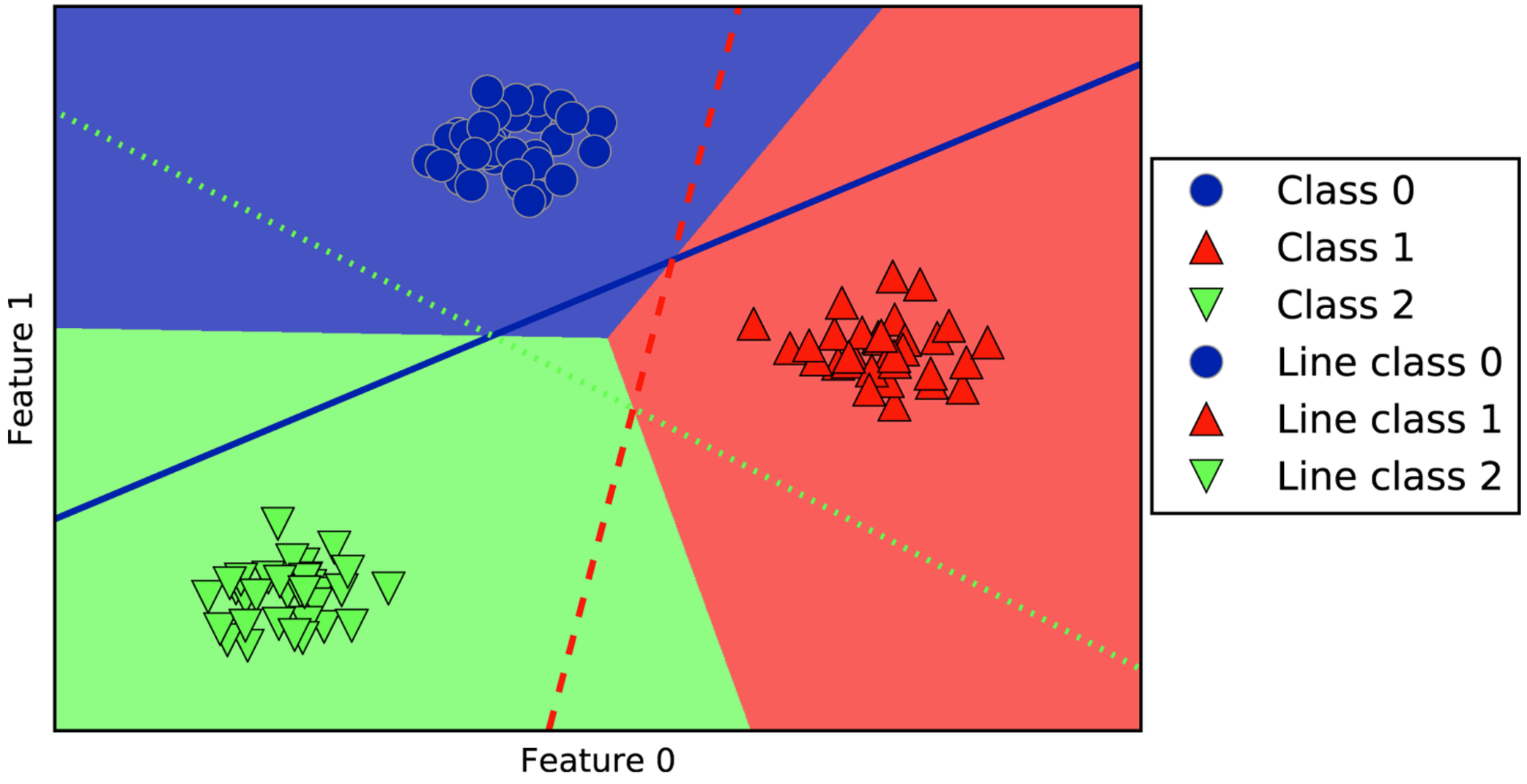


Figure 2-21. Multiclass decision boundaries derived from the three one-vs.-rest classifiers

Jupyter Notebook 02-supervised-learning .ipynb [49]

Linear Models Overview

- **Assumptions**
  - Linear relationship between input and output
  - No noise in input or output
  - Independence (lack of correlation in input)
  - Gaussian (normally) distributed data

- **Best Practices**
  - Scale inputs

Linear Models Overview

- **Parameters**
  - Regularization parameter: alpha (regression) or C (classification)
  - Type of regularization (L1 or L2)
- **Strengths**
  - Fast to train (good for large datasets)
  - Fast at prediction
  - Easy to understand the algorithm
  - Work well even when there are a large number of features compared to the number of samples
- **Weaknesses**
  - Not so easy to interpret a model's coefficients
  - Generalization may be poor with small number of features