LONDON METROPOLITAN UNIVERSITY

Machine Learning
CS7052
Lecture 4, Linear Models

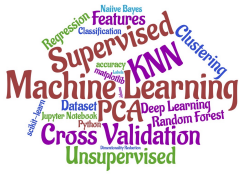
Dr. Elaheh Homayounvala

Autumn, 2023-2024, week 4

Outline of today's lecture

- Review last week
  - K-NN Classification and Regression
  - Overfitting and underfitting

- Supervised learning
  - Linear models, Muller and Guido's book page 47-70
    - Linear Regression
    - Linear Models for classification



Review last week

k-Nearest Neighbours

Overfitting and underfitting

K-NN Classification

- To make a prediction for a new data point, the algorithm finds the closest data points in the training dataset—its "nearest neighbours."
- Simplest version, K=1, K-NN considers exactly **one** nearest neighbour
- Which is the **closest** training data point to the point we want to make a prediction for.
- Reference: Muller & Guido's book page 37

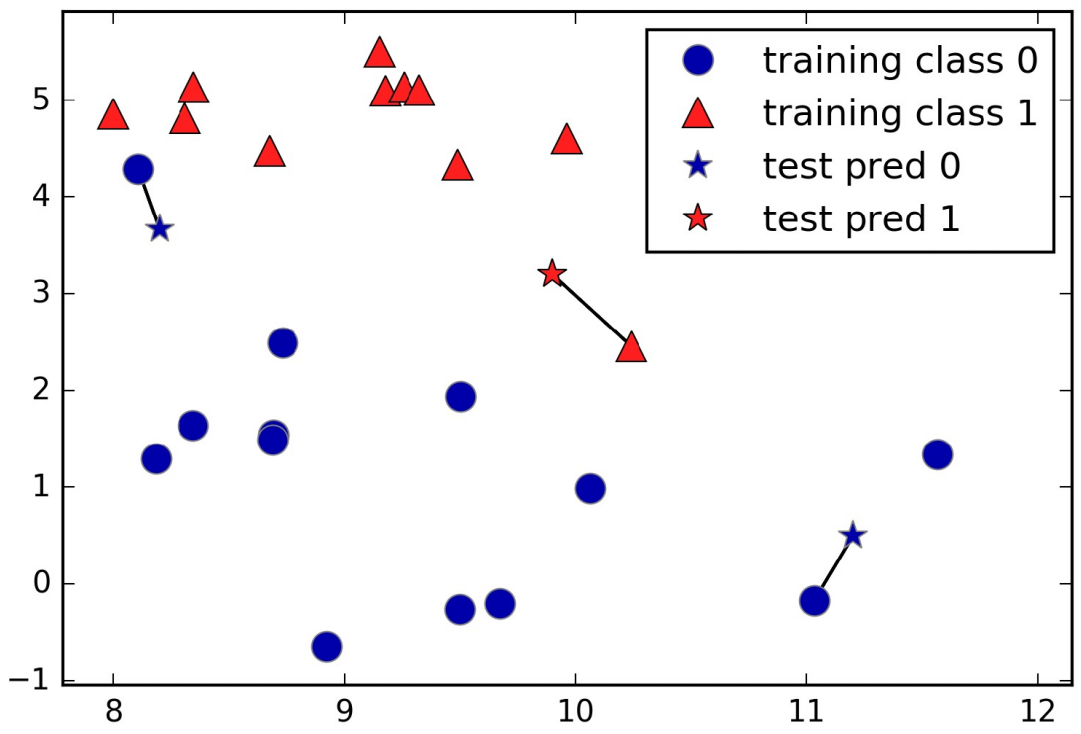
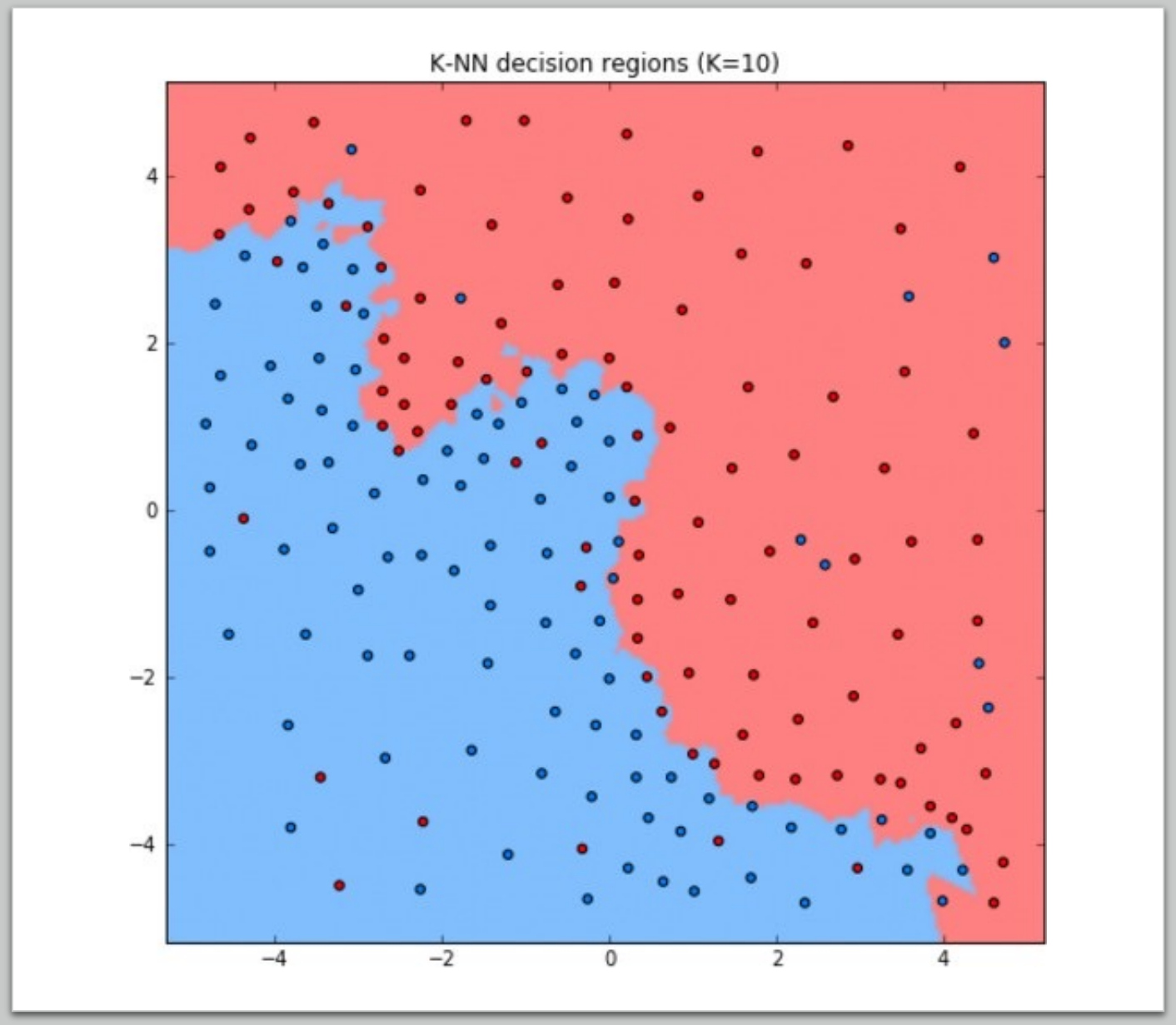


Figure 2-4. Predictions made by the one-nearest-neighbor model on the forge dataset

Example: K-NN Classification

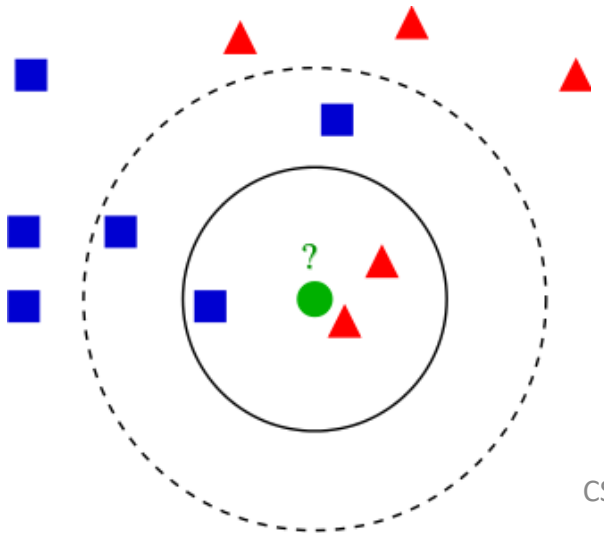
- Decision boundaries or Decision regions
- When k = 10



k-Nearest Neighbours Algorithm

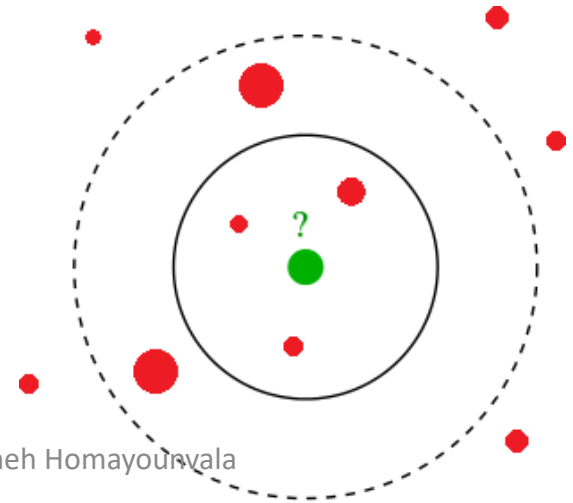
Classification:

1. Find k closest objects to the predicted object x in the training set.
2. Associate x the most frequent class among its k neighbours.



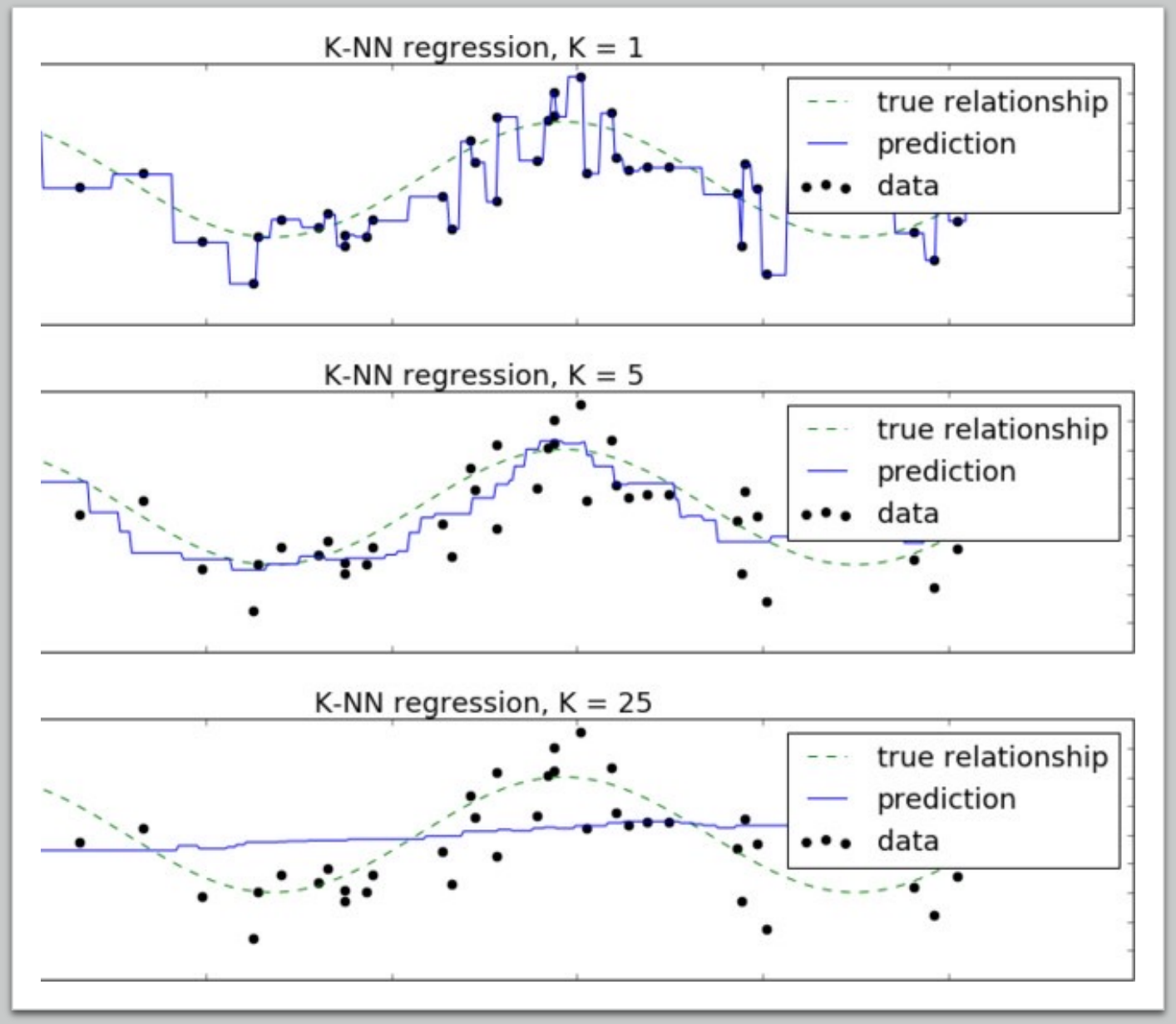
Regression:

1. Find k closest objects to the predicted object x in the training set.
2. Associate x average output of its k neighbours.



K-NN Regression

- Can we predict numerical values using k-NN Algorithm?



Parameters

- There are two parameters in k-NN:
  - The number of neighbours
  - How you measure distance between data points
    - Euclidian distance
    - What are the other ways to measure distance?

Overfitting and Underfitting

- Choosing too simple a model is called underfitting.
  - It is not even good on train data

- Overfitting occurs when you fit a model too closely to the particularities of the training set
  - High accuracy on train data but not on test data

Model Complexity vs. Accuracy

- Underfitting
- Overfitting

- The sweet spot

- Reference: Muller & Guido's book page 31

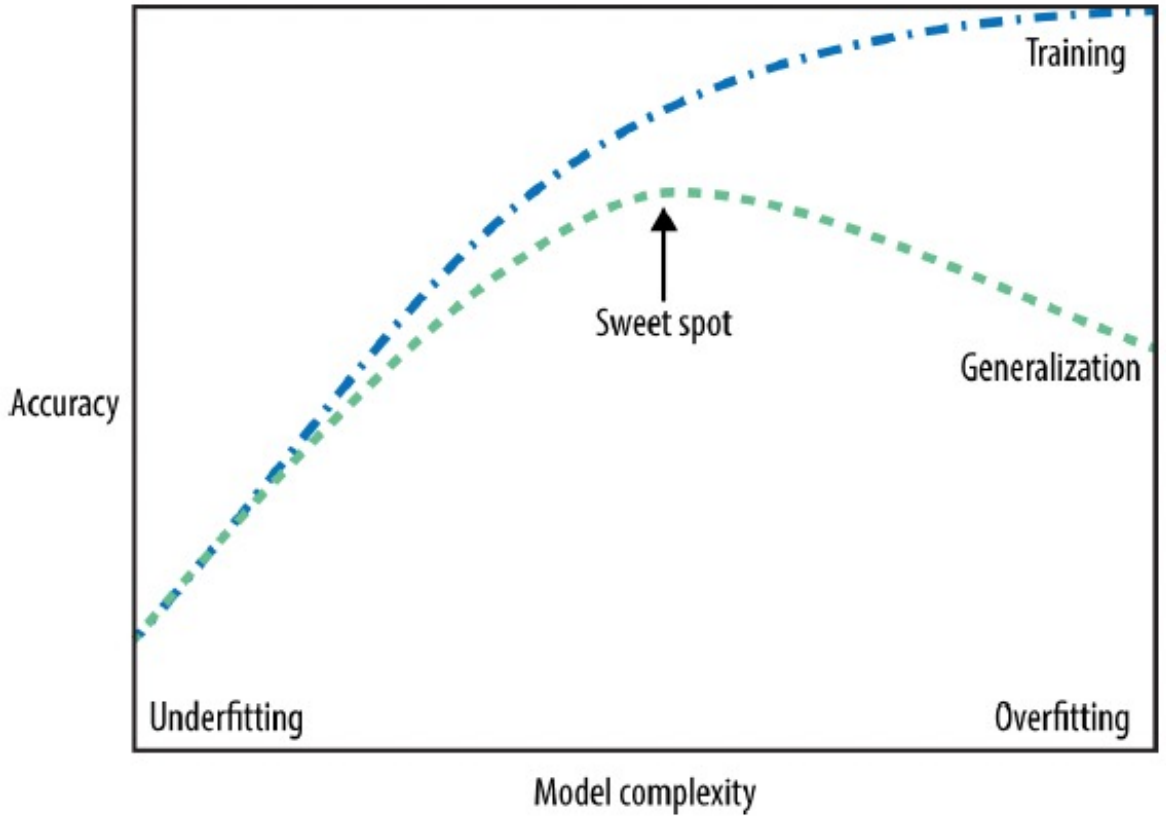
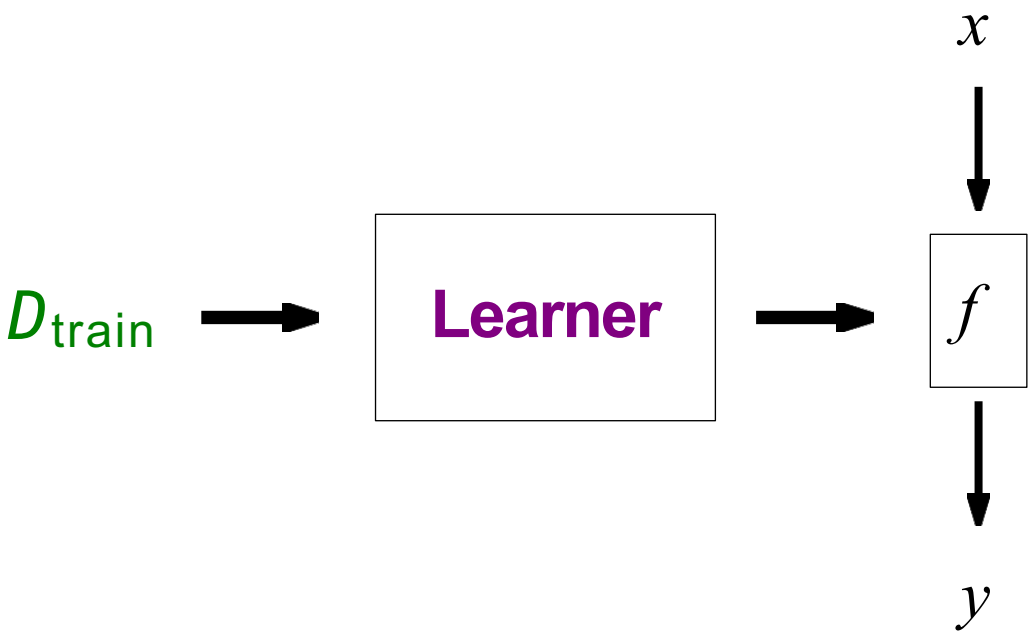


Figure 2-1. Trade-off of model complexity against training and test accuracy

Supervised Learning Linear Models

Linear Regression

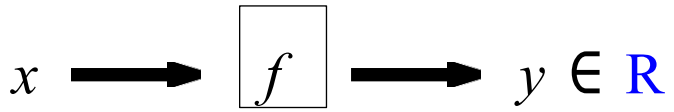
Supervised Learning



Types of prediction tasks

Binary classification (e.g., email ⇒ spam/not spam):

Regression (e.g., location, year ⇒ housing price):



Linear Regression

- Linear models make a prediction using a linear function of the input features

- Source: page 48, Muller and Guidio's book

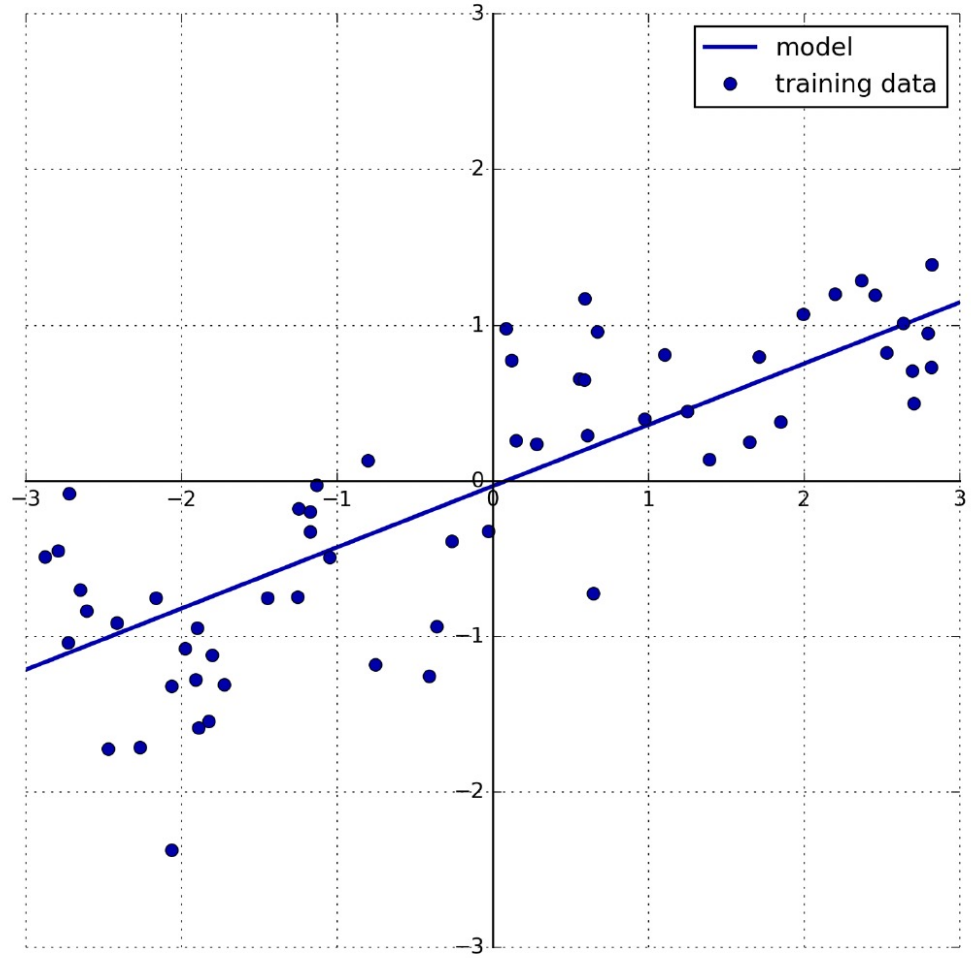


Figure 2-11. Predictions of a linear model on the wave dataset

Linear Models of Regression

$$\hat{y} = w[0] * x[0] + b$$

- w[0] is the slope
- b is the y-axis offset or the intercept

$$\hat{y} = a * x[0] + b$$ might look more familiar

- In case of more than one feature:

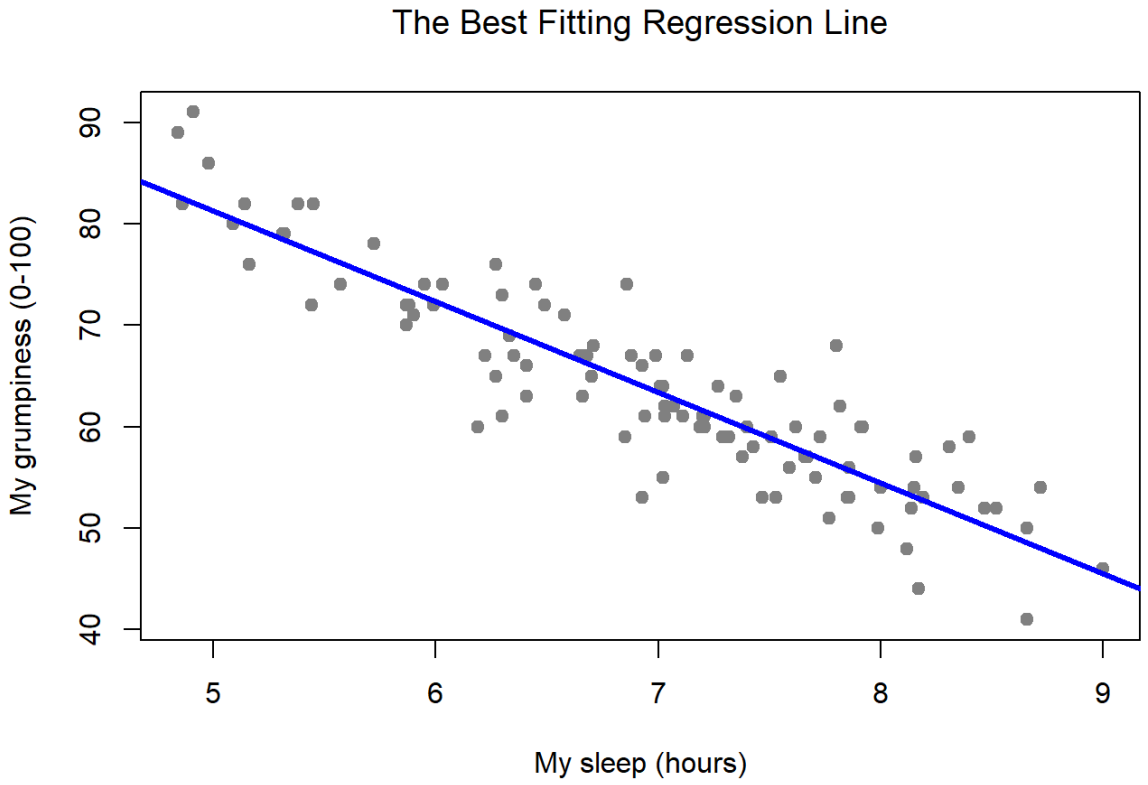
$$\hat{y} = w[0] * x[0] + w[1] * x[1] + ... + w[p] * x[p] + b$$

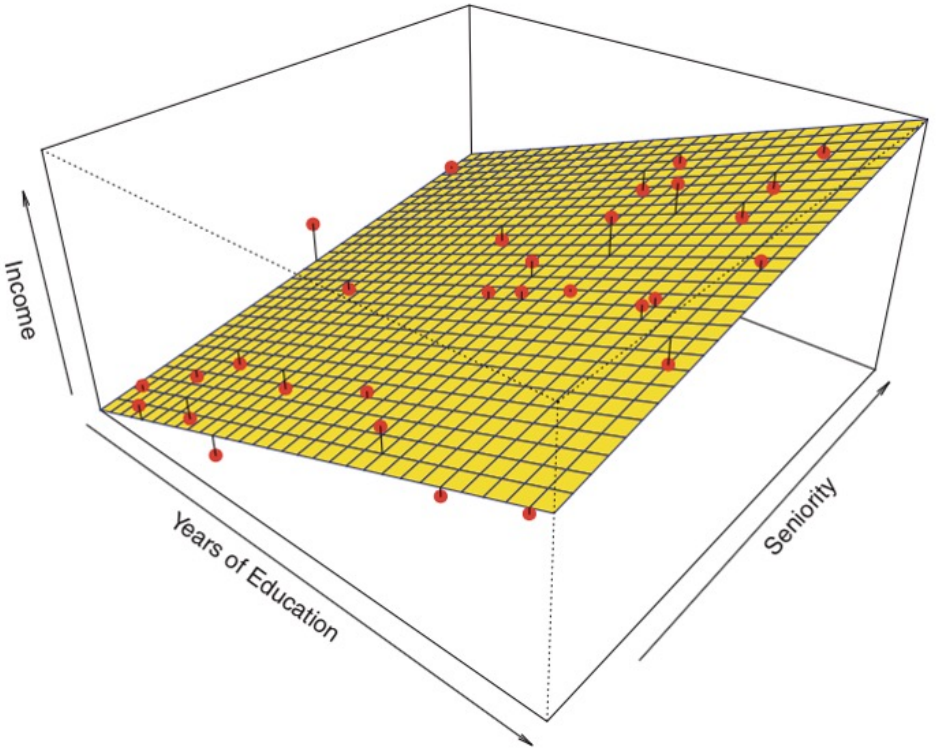
Linear Models for Regression

Can be characterized as regression models for which the prediction is:

- a line for a single feature,
- a plane when using two features,
- a hyperplane in higher dimensions (that is, when using more features).

Regression Line, Regression Plane





Linear Regression

- Which line is a better fit to the training data?
- How would you know?

- Any suggestion?

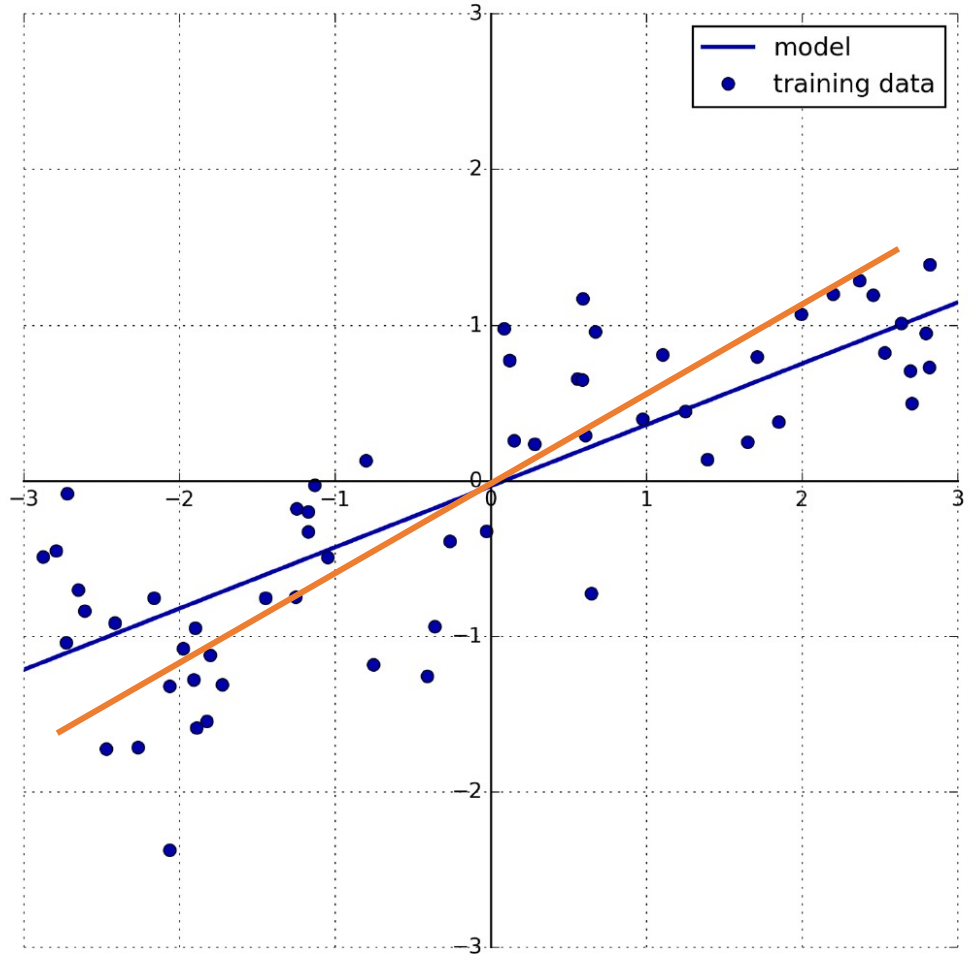


Figure 2-11. Predictions of a linear model on the wave dataset

Linear Regression

- Which line is a better fit to the training data?

- Cost function
- Residual Sum of Squares

$$RSS = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

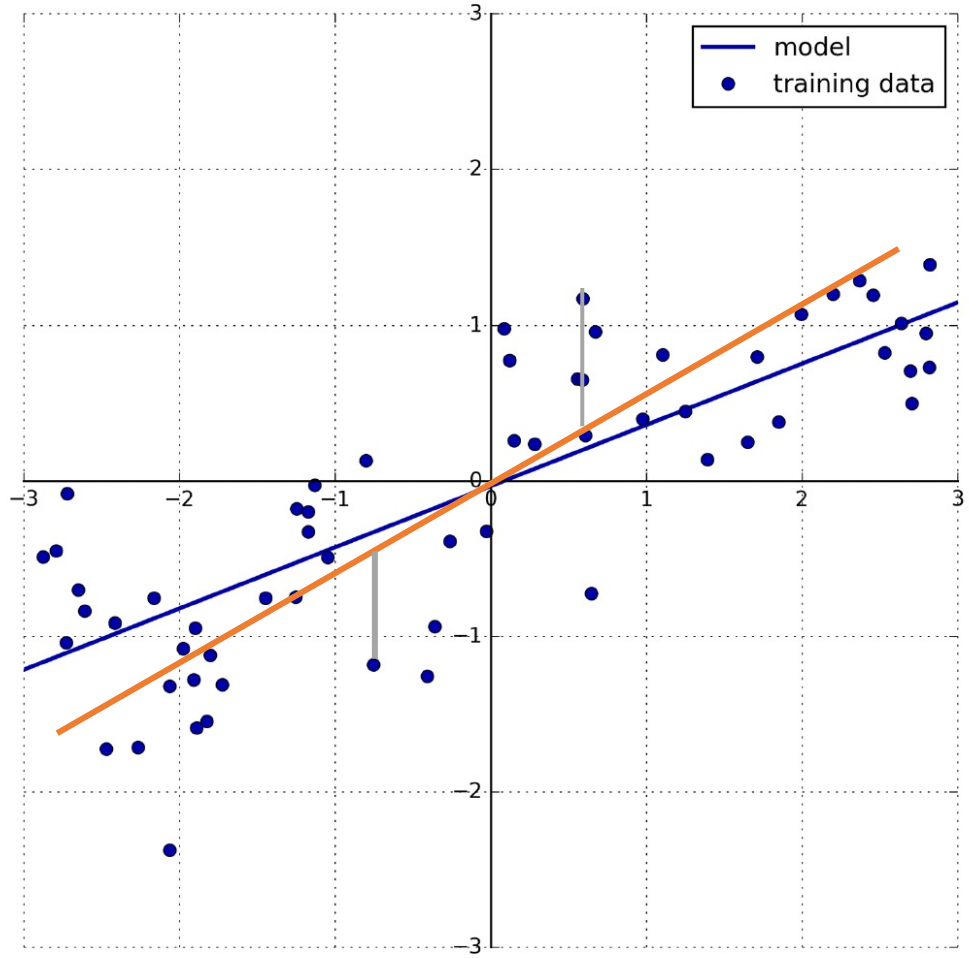


Figure 2-11. Predictions of a linear model on the wave dataset

Linear Regression and Cost function

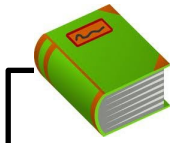
$$\hat{y} = w[0] * x[0] + w[1] * x[1] + \ldots + w[p] * x[p] + b$$

- Linear regression looks for optimizing *w* and *b* such that it minimizes the cost function
- The cost function can be written as:

$$\sum_{i=1}^{M} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{M} \left(y_i - \sum_{j=0}^{p} w_j \times x_{ij} \right)^2$$

- The data-set has M instances and p features

Cost or Loss function



Definition: loss function

A loss function Loss(x, y, **w**) quantifies how unhappy you would be if you used **w** to make a prediction on x when the correct output is y. It is the object we want to minimize.

Linear Regression, underfitting and overfitting

- If we have very few features on a data-set and the score is poor for both training and test set, then it's a problem of underfitting.

- if we have large number of features and test score is relatively poor than the training score then it's the problem of overfitting

Linear Regression

- Simple linear Regression (Ordinary Least squares)
- Ridge Regression
- Lasso Regression (least absolute shrinkage and selection operator)

Ridge Regression

- Ridge regression is a linear regression model based on least squares.
- In ridge regression, the coefficients (w) are chosen not only so that they predict well on the training data, but also to fit an additional constraint.
- We also want the magnitude of coefficients to be as small as possible; in other words, all entries of w should be close to zero.

Ridge Regression

- In Ridge regression, the cost function is altered by adding a penalty equivalent to square of the magnitude of the coefficients.

$$\sum_{i=1}^{M} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{M} \left(y_i - \sum_{j=0}^{p} w_j \times x_{ij} \right)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

Cost function for ridge regression

Trade-off

The Simplicity of the Model (Near-zero Coefficients)

vs.

Performance of the Model on the Training Set

- How much importance the model places on simplicity versus training set performance can be specified by the user
- using **the alpha** parameter.

Regularisation and Ridge Regression

- Regularization means explicitly restricting a model to avoid overfitting.
- L2 regularisation is used by ridge regression.
- Each feature should have as little effect on the outcome as possible (having a small slope), while still predicting well
- This constraint is an example of regularisation.

Different values of alpha

- Muller and Guido's book, page 53

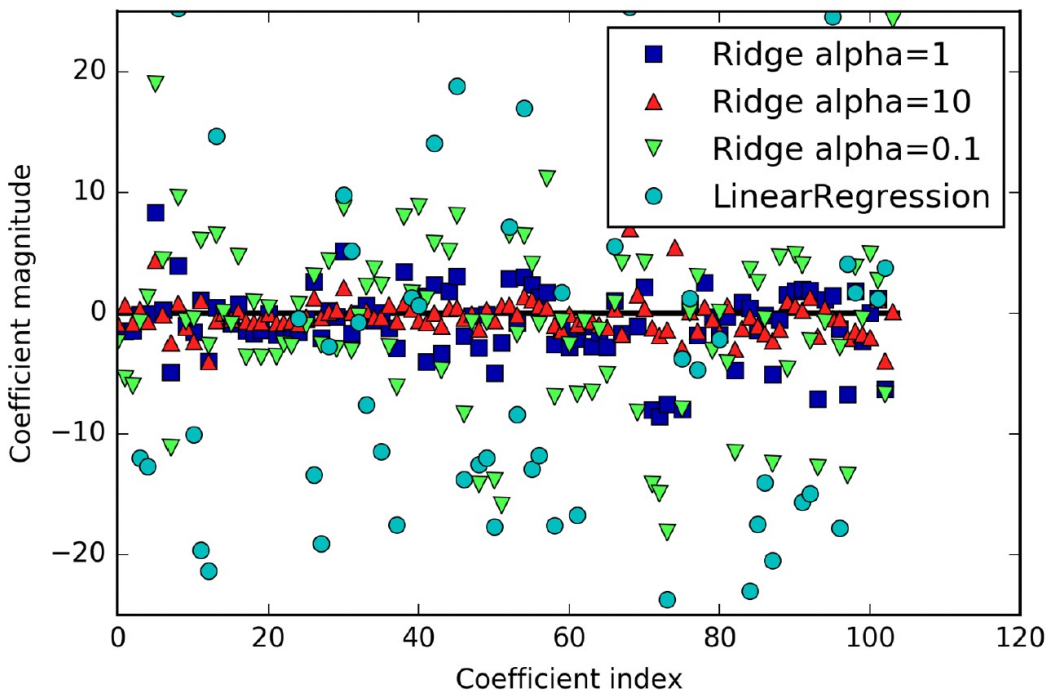


Figure 2-12. Comparing coefficient magnitudes for ridge regression with different values of alpha and linear regression

Different values of training set size

- Muller and Guido's book, page 54

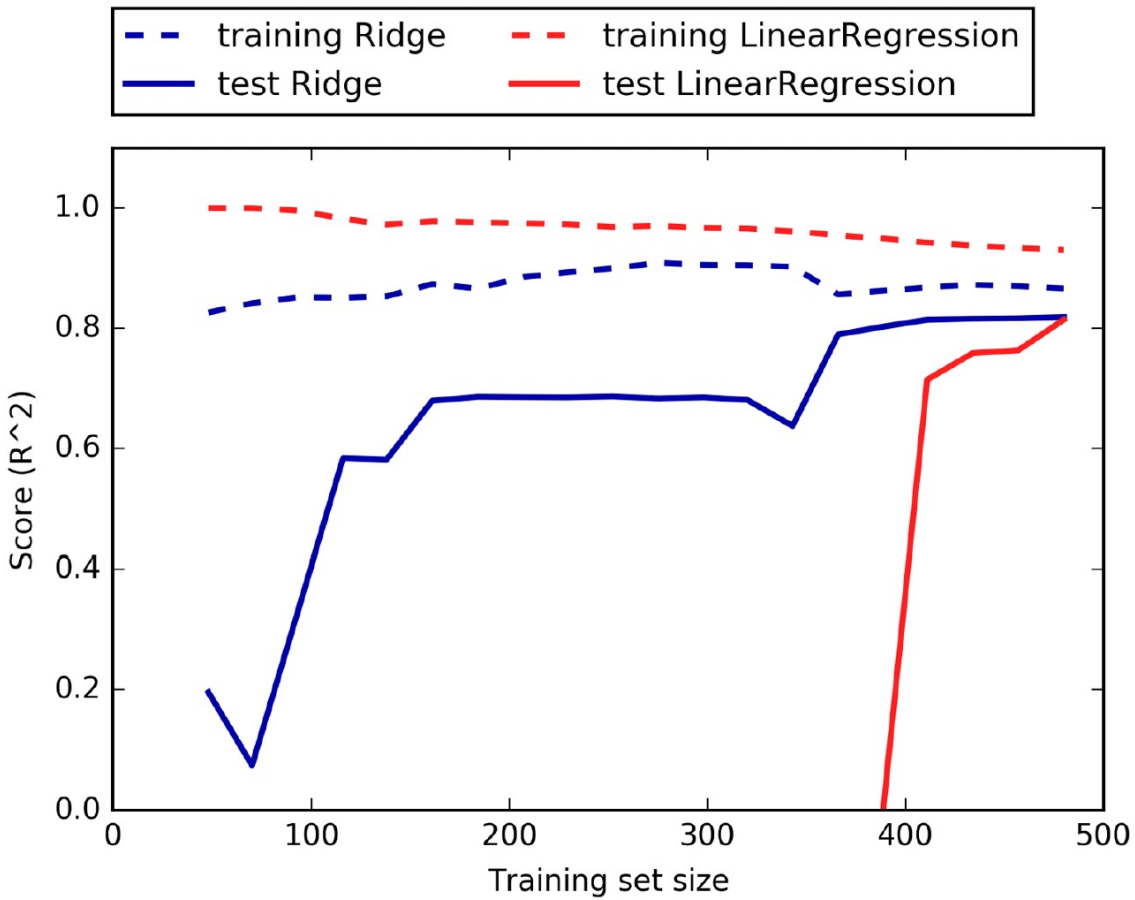


Figure 2-13. Learning curves for ridge regression and linear regression on the Boston Housing dataset

Lasso Regression

- Is a linear regression model
- Similar to Ridge, Lasso restricts coefficients to be close to zero
- L1 regularization
- Some coefficients are exactly zero.
- Some features are entirely ignored by the model
- makes a model easier to interpret
- Cost function in Lasso:

$$\sum_{i=1}^{M}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p} w_j \times x_{ij}\right)^2 + \lambda \sum_{j=0}^{p} |w_j|$$

Parul Pandey

Just like Ridge regression cost function, for lambda =0, the equation above

Different values of alpha

- Muller and Guido's book, page 57

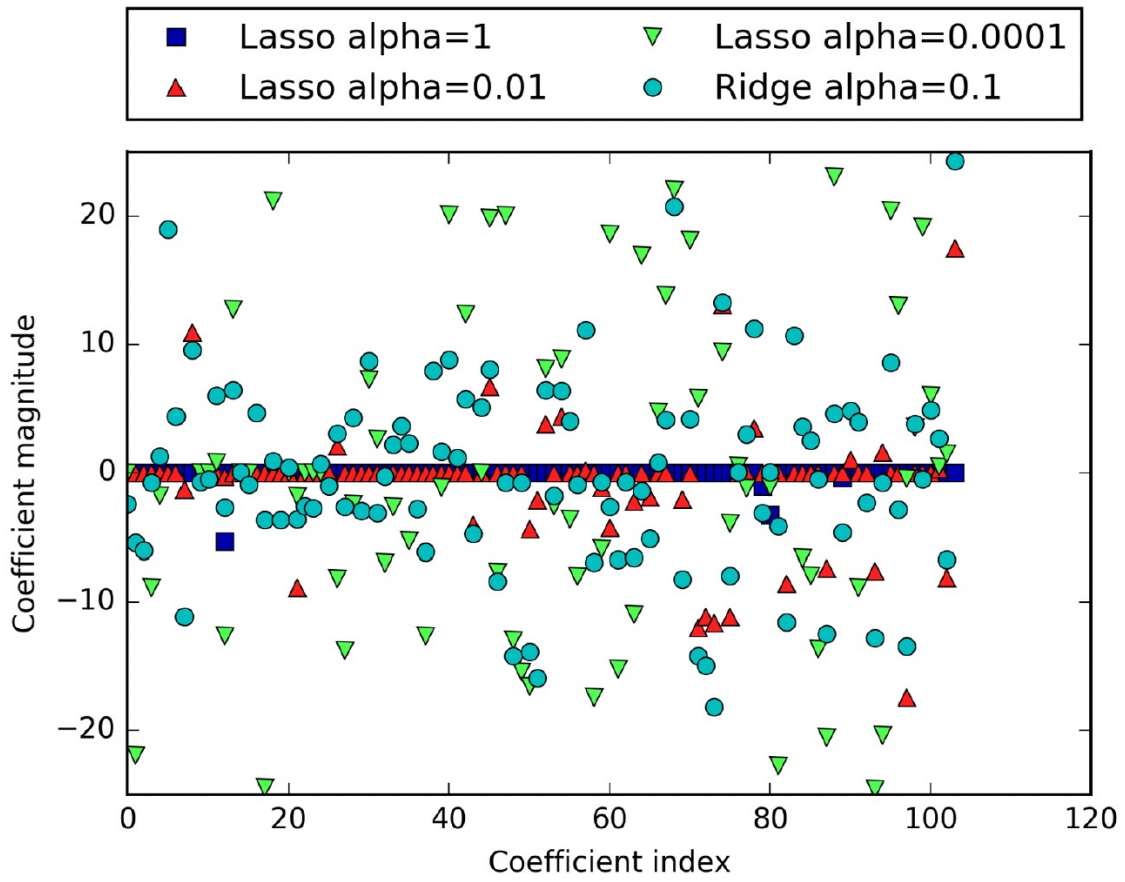


Figure 2-14. Comparing coefficient magnitudes for lasso regression with different values of alpha and ridge regression

Ridge and Lasso compared

- Ridge regression is usually the first choice
- Lasso is a better choice, if you have many features and expect only a few of them to be important
- Lasso is a better choice if you would like to have a model that is easy to interpret
- Lasso will provide a model that is easier to understand

Supervised Learning Linear Models

Linear Models for Classification

Classification Application example spam email classification

Input: x = email message

From: pliang@cs.stanford.edu Date: September 27, 2017 Subject: CS221 announcement Hello students, I've attached the answers to homework 1...	From: a9k62n@hotmail.com Date: September 27, 2017 Subject: URGENT Dear Sir or maDam: my friend left sum of 10m dollars...

Output: $y \in \{\text{spam}, \text{not-spam}\}$

Objective: obtain a **predictor** f

Linear Classification

- $\hat{y} = w[0] * x[0] + w[1] * x[1] + ... + w[p] * x[p] + b > 0$
- In classification, we threshold the weighted sum of features at zero

$$\text{Class } +1, \quad \text{if } w[0] * x[0] + w[1] * x[1] + ... + w[p] * x[p] + b > 0$$

$$\text{Class } -1, \quad \text{if } w[0] * x[0] + w[1] * x[1] + ... + w[p] * x[p] + b < 0$$

Linear Classification

- Linear models for classification, the decision boundary is a linear function of the input.
- A (binary) linear classifier is a classifier that separates two classes using
  - a line
  - a plane
  - or a hyperplane.

Linear Classification Algorithms

- There are many algorithms for learning linear models.

- These algorithms all differ in the following two ways:
  - The way in which they measure how well a particular combination of coefficients and intercept fits the training data
  - If and what kind of regularisation they use

Linear Classification Algorithms

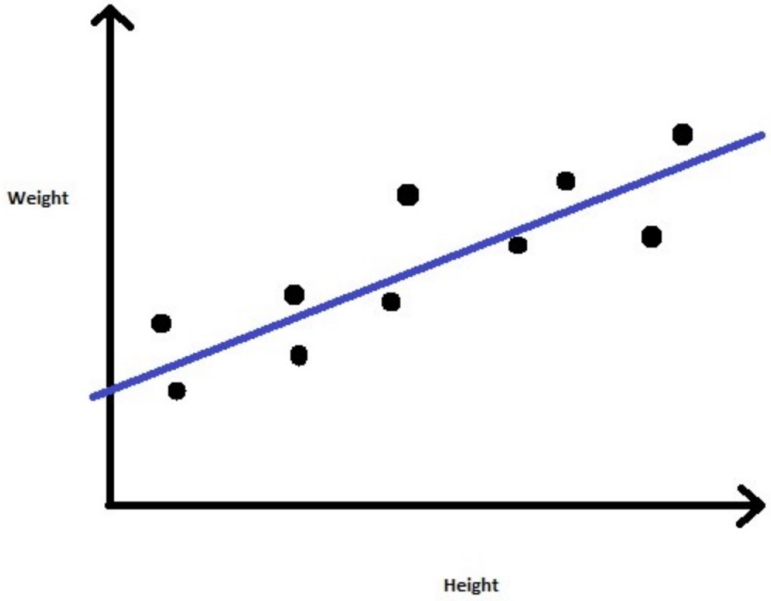
- Logistic Regression
- Linear Support Vector Machines (Linear SVMs)

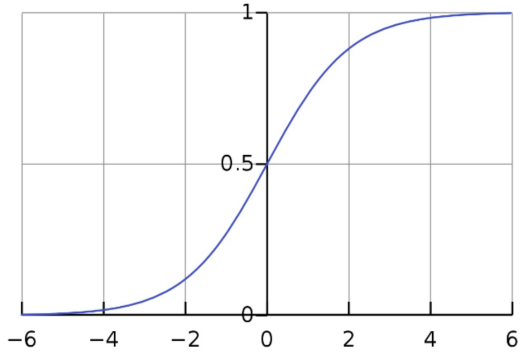
- Both use L2 regularisation similar to Ridge

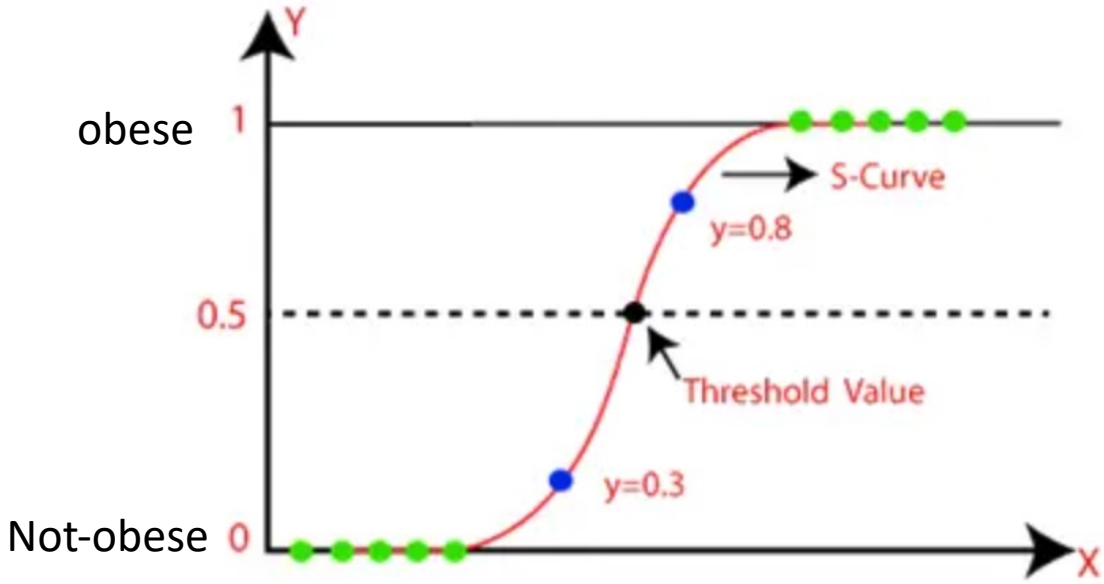
Logistic Regression

- We start with linear regression
- This regression line is highly susceptible to outliers, it will not do a good job in classifying two classes
- To get a better classification, we will feed the output values from the regression line to the sigmoid function.
- The sigmoid function returns the probability for each output value from the regression line.
- Now based on a predefined threshold value, we can easily classify the output into two classes Obese or Not-Obese.

Logistic Regression, example





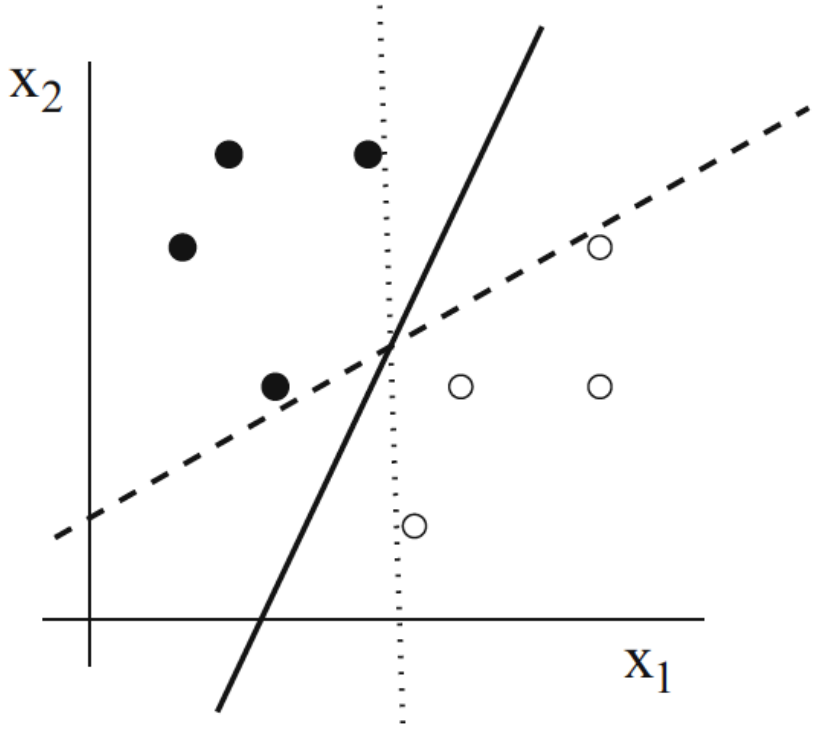


Linear Support Vector Machine

- We find the points closest to the line from both the classes.
- These points are called support vectors.
- Now, we compute the distance between the line and the support vectors.
- This distance is called the margin.
- Our goal is to maximize the margin.
- The hyperplane for which the margin is maximum is the optimal hyperplane

Linear Support Vector Machine, example

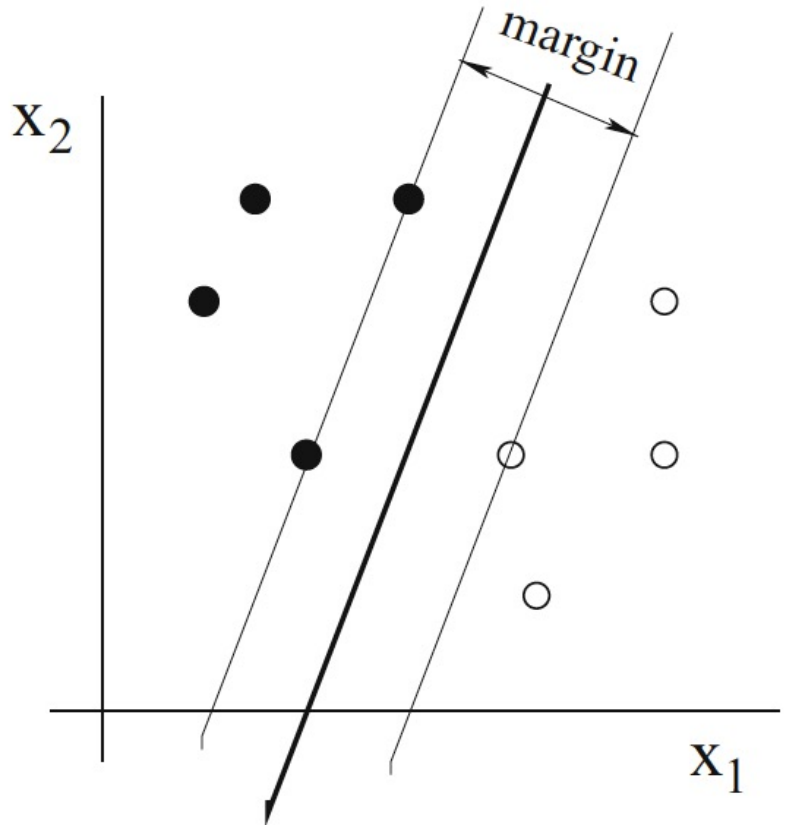
Fig. 4.8 Linearly separable classes can be separated in infinitely many different ways. Question is, which of the classifiers that are perfect on the training set will do best on future data



Source: An introduction to Machine Learning, Miroslav Kobat page 85

Linear Support Vector Machine, example

Fig. 4.9 The technique of the *support vector machine* looks for a separating hyperplane that has the maximum *margin*



Source: An introduction to Machine Learning, Miroslav Kobat page 86

Linear SVM and Logistic Regression Decision boundaries

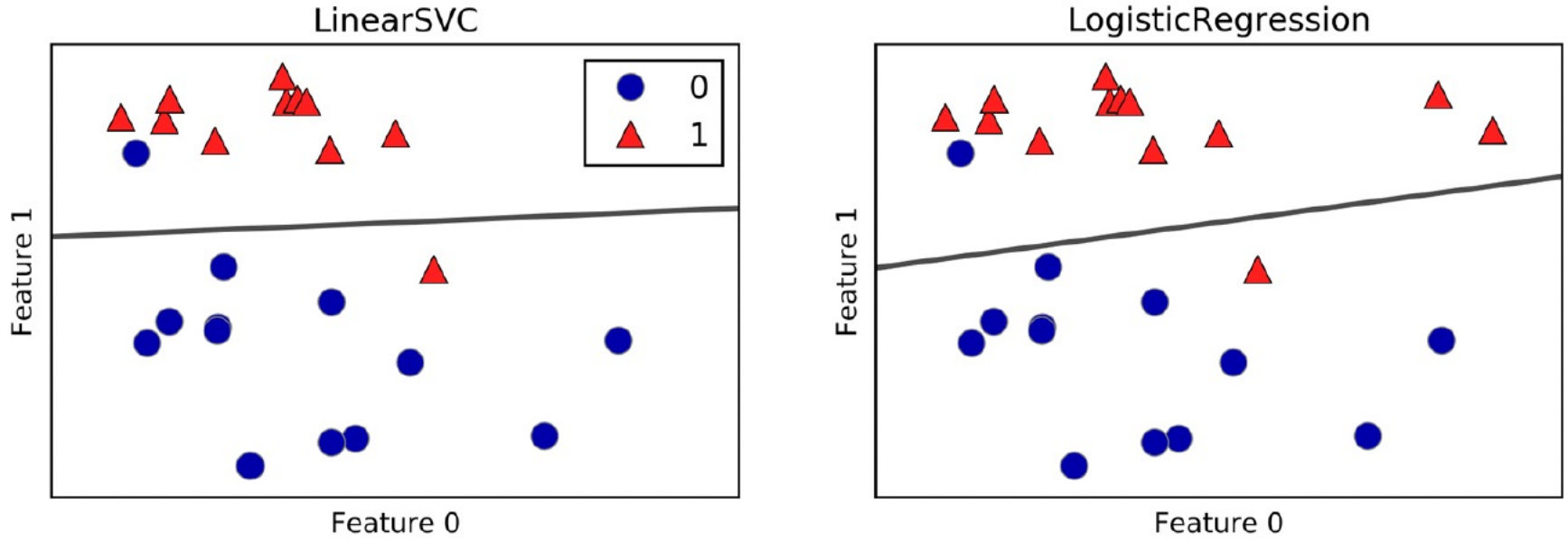


Figure 2-15. Decision boundaries of a linear SVM and logistic regression on the forge dataset with the default parameters

Source: Muller and Guido's book, page 59

The strength of regularization

- C parameter
- Higher values of C
  - Less regularisation
  - Logistic Regression and Linear SVM try to fit the training set as best as possible
  - Stresses the importance that each individual data point be classified correctly.
- Low values of C
  - The models put more emphasis on finding a coefficient vector (w) that is close to zero.
  - Cause the algorithms to try to adjust to the “majority” of data points

Different C values for Linear SVM

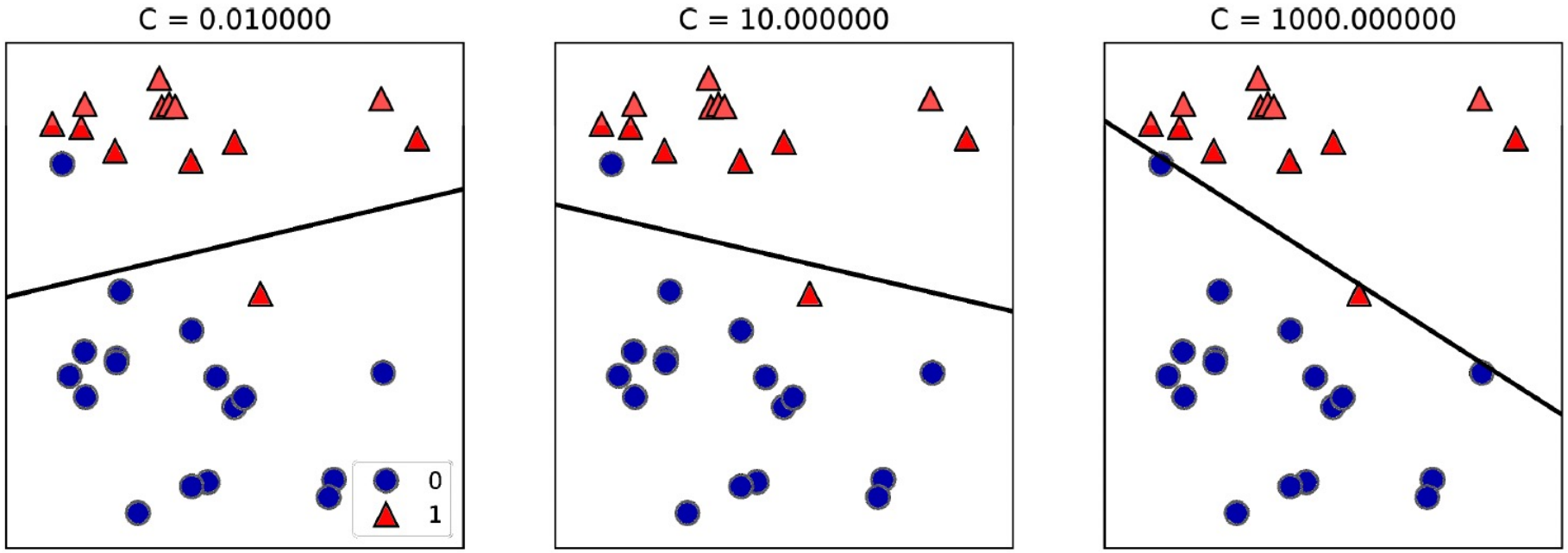


Figure 2-16. Decision boundaries of a linear SVM on the forge dataset for different values of C

Which one is likely overfitting?

Source: Muller and Guido's book, page 60

High dimensions

- In high dimensions, linear models for classification become very powerful

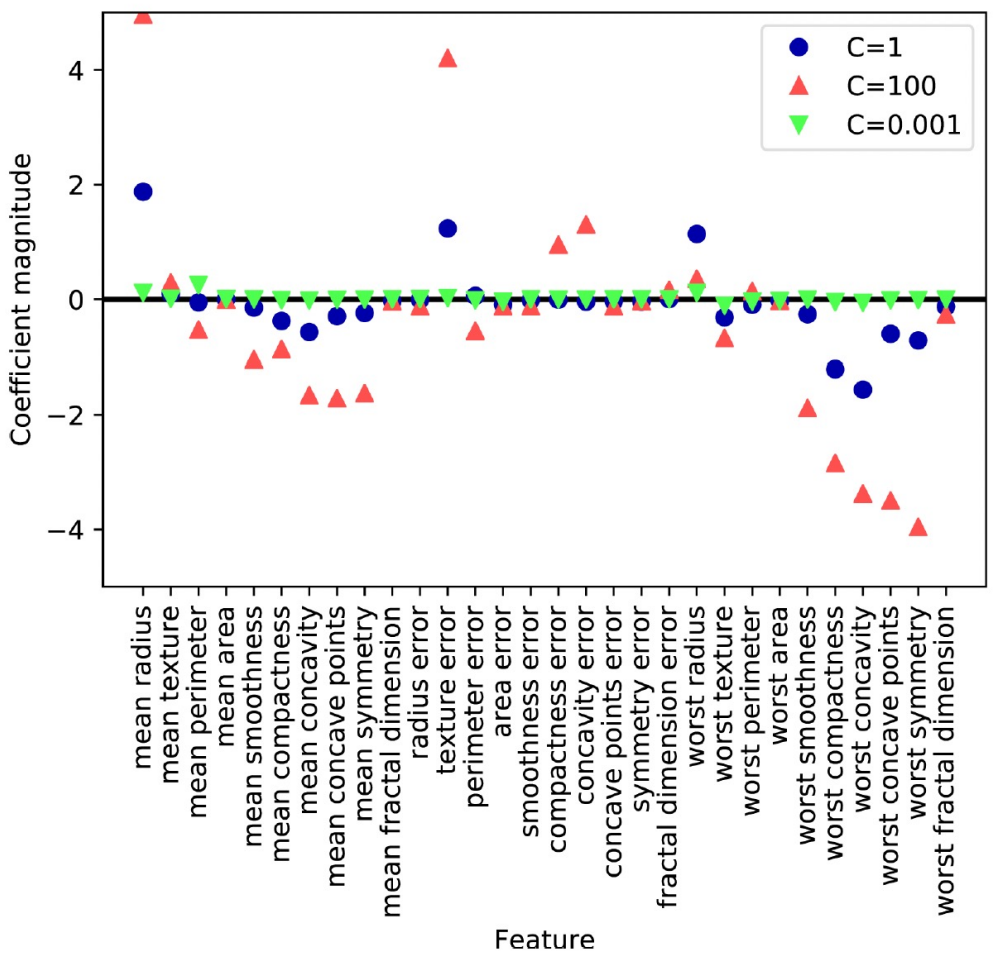


Figure 2-17. Coefficients learned by logistic regression on the Breast Cancer dataset for different values of C

Source: Muller and Guido's book, page 63

- L1 regularization
- More interpretable model

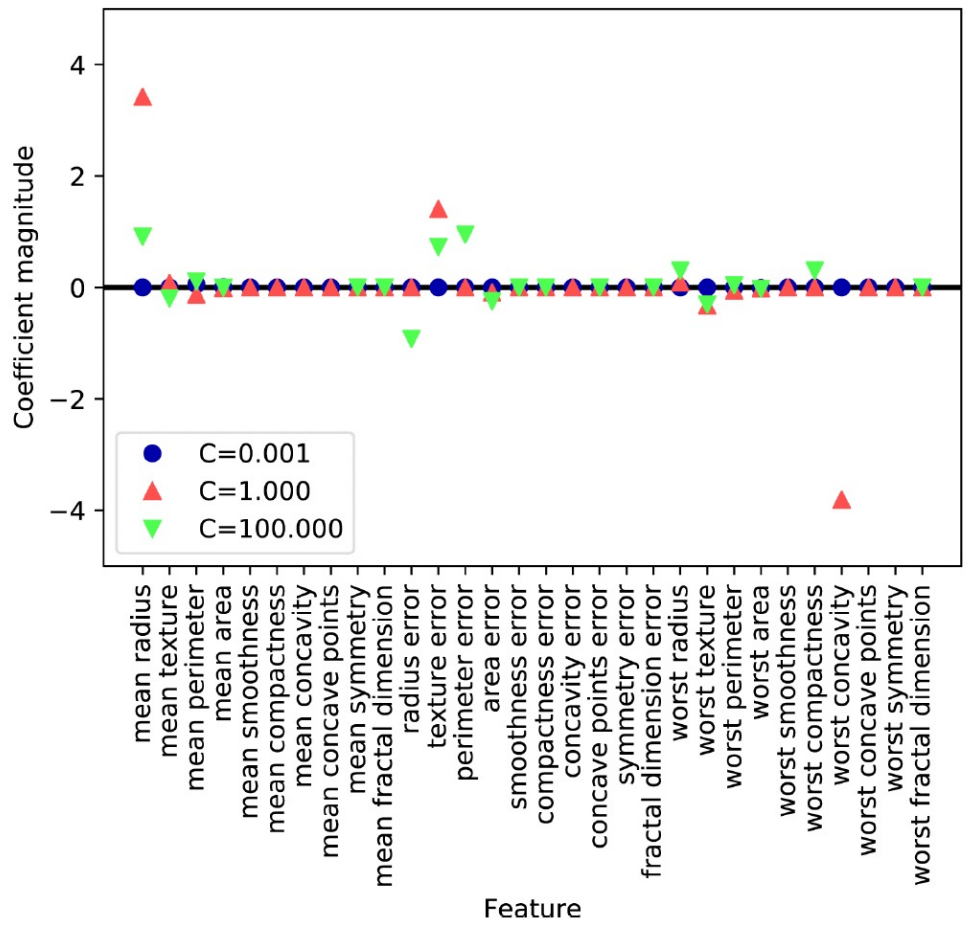


Figure 2-18. Coefficients learned by logistic regression with L1 penalty on the Breast Cancer dataset for different values of C

Source: Muller and Guido's book, page 65

Linear Models for Multi-class Classification

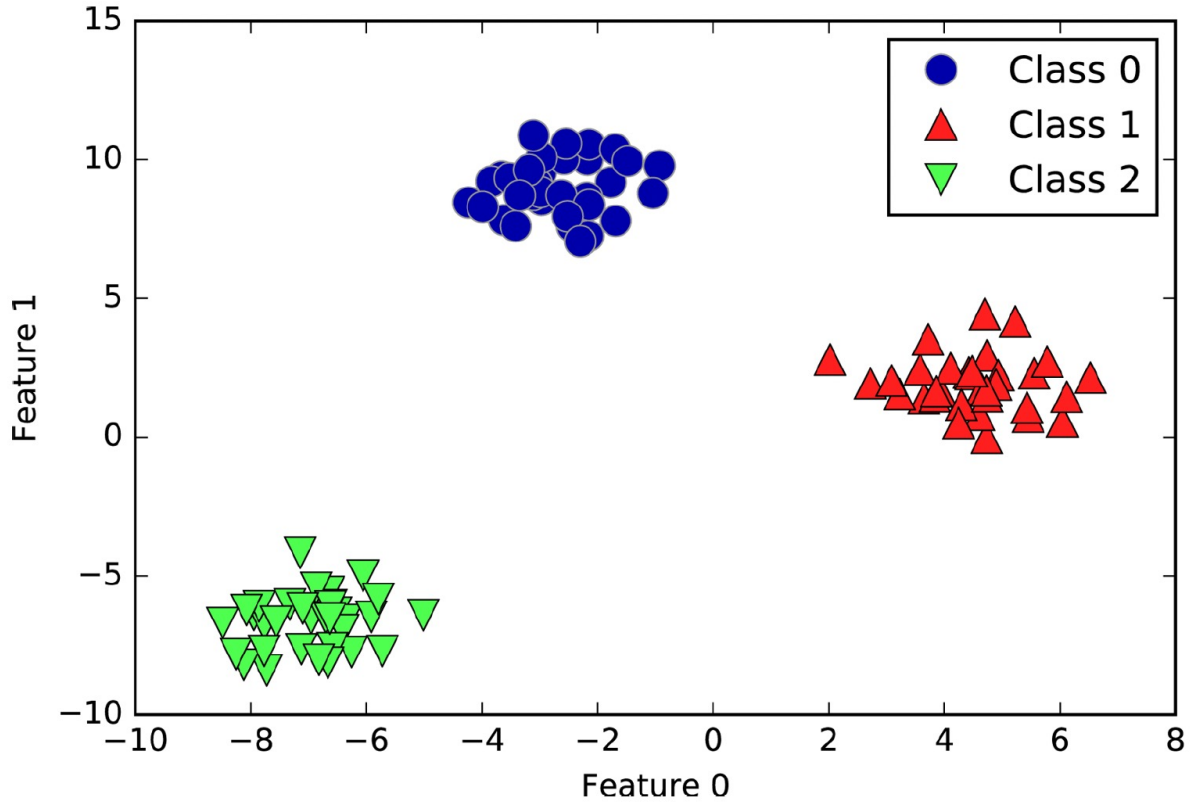


Figure 2-19. Two-dimensional toy dataset containing three classes

Source: Muller and Guido's book, page 66

Linear Models for Multi-class Classification

- One-vs.-rest approach
- Multiclass logistic regression

One-vs.-rest approach

- A binary model is learned for each class that tries to separate that class from all of the other classes
- Resulting in as many binary models as there are classes
- To make a prediction, all binary classifiers are run on a test point.
- The classifier that has the highest score on its single class “wins,” and this class label is returned as the prediction.

One-vs.-rest approach, example

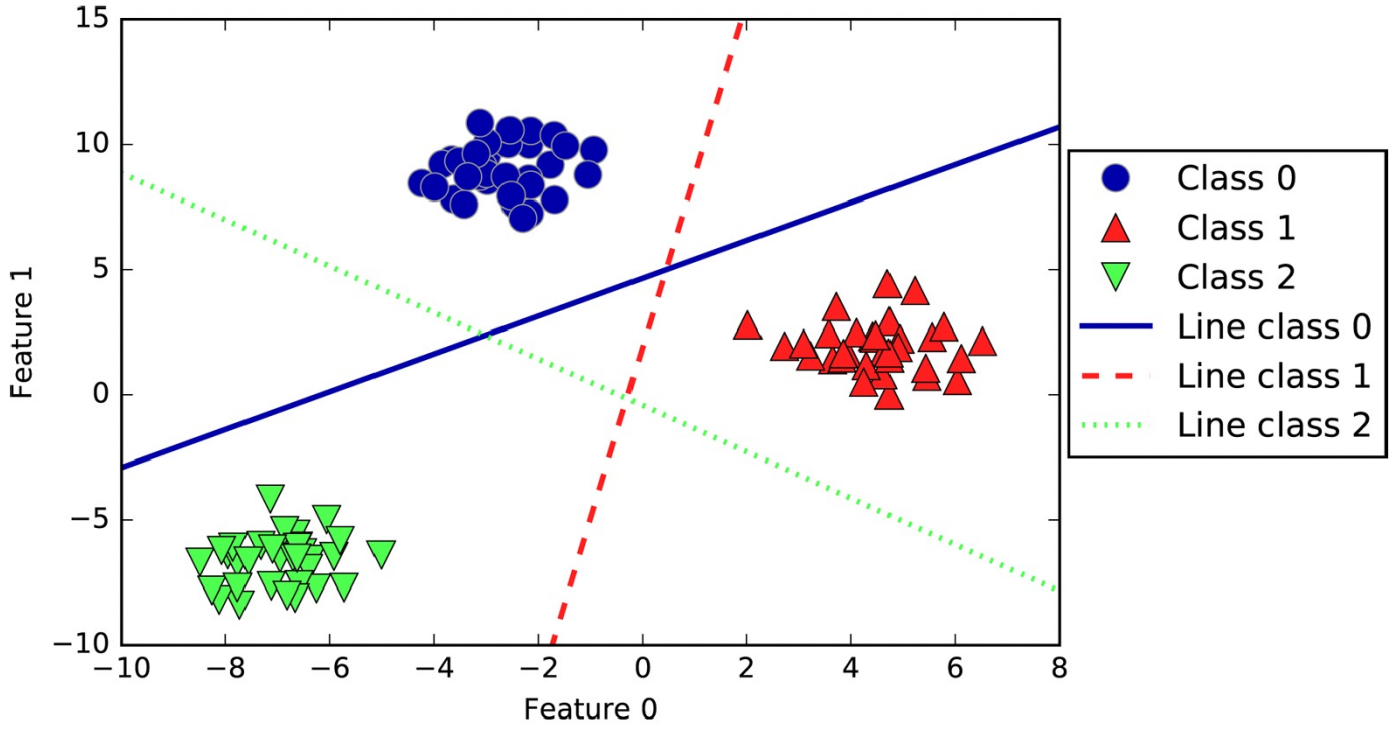


Figure 2-20. Decision boundaries learned by the three one-vs.-rest classifiers

Source: Muller and Guido's book, page 68

Multi-class Logistic Regression

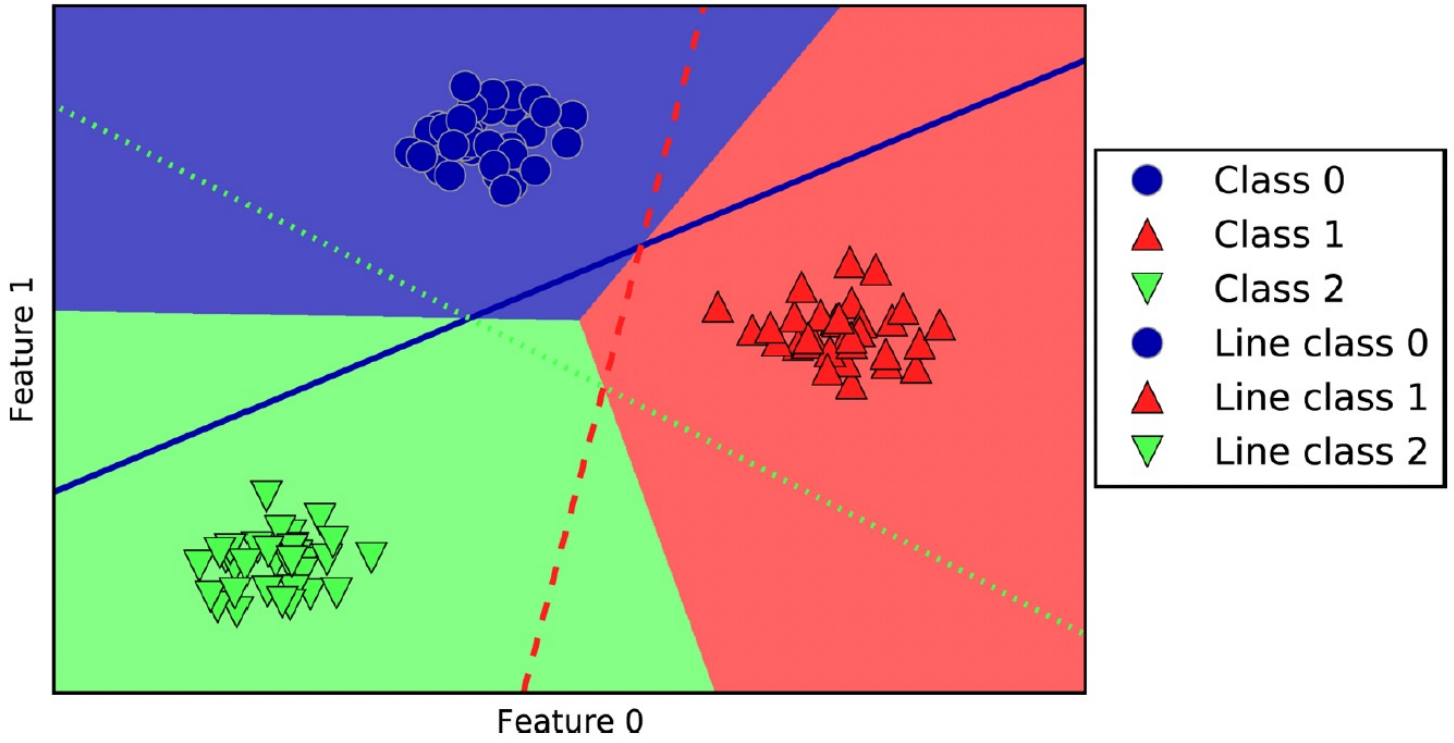
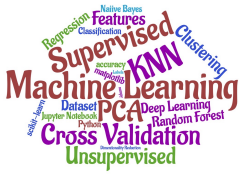


Figure 2-21. Multiclass decision boundaries derived from the three one-vs.-rest classifiers

Source: Muller and Guido's book, page 69



Strength, weaknesses and parameters

Of linear model

Parameters, alpha and C

- Alpha in regression models
- C in Linear SVM and Logistic Regression

- Large values for alpha or small values for C mean simple models

Parameters, L1 or L2

- Use L1 regularization
  - If you assume that only a few of your features are actually important,
  - if interpretability of the model is important

- L2 regularization
  - default

Strengths and weaknesses

- Fast in training
- Fast in prediction
- Scale well to very large datasets
- Work well with sparse data
- Relatively easy to understand how a prediction is made
- Not entirely clear why coefficients are the way they are
- Perform well when the number of features is large compared to the number of samples.
- in lower-dimensional spaces, other models might yield better generalization performance

Last weeks workshops

- Iris species

Iris dataset

- source: Muller and Guido's book, page 20

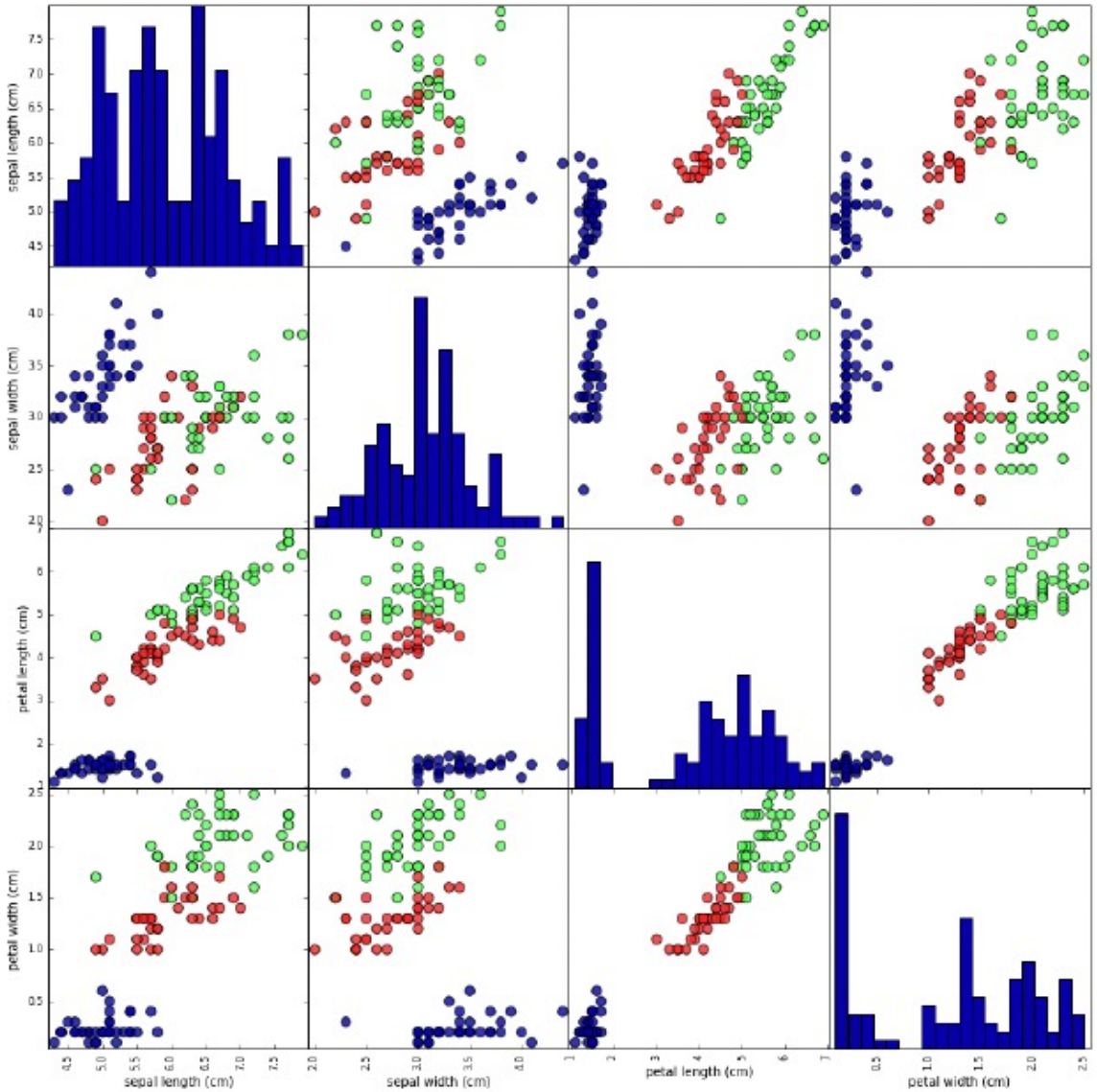


Figure 1-3. Pair plot of the Iris dataset, colored by class label