Lecture 009

Support vector machines

Edward Rubin
03 March 2020

Admin

Today

- *Mini-survey* What are you missing?
- *Results* In-class competition
- *Topic* Support vector machines

Upcoming

Readings

- *Today* *ISL* Ch. 9
- *Next* *100ML* Ch. 6

Project Project updates/questions?

In-class competition

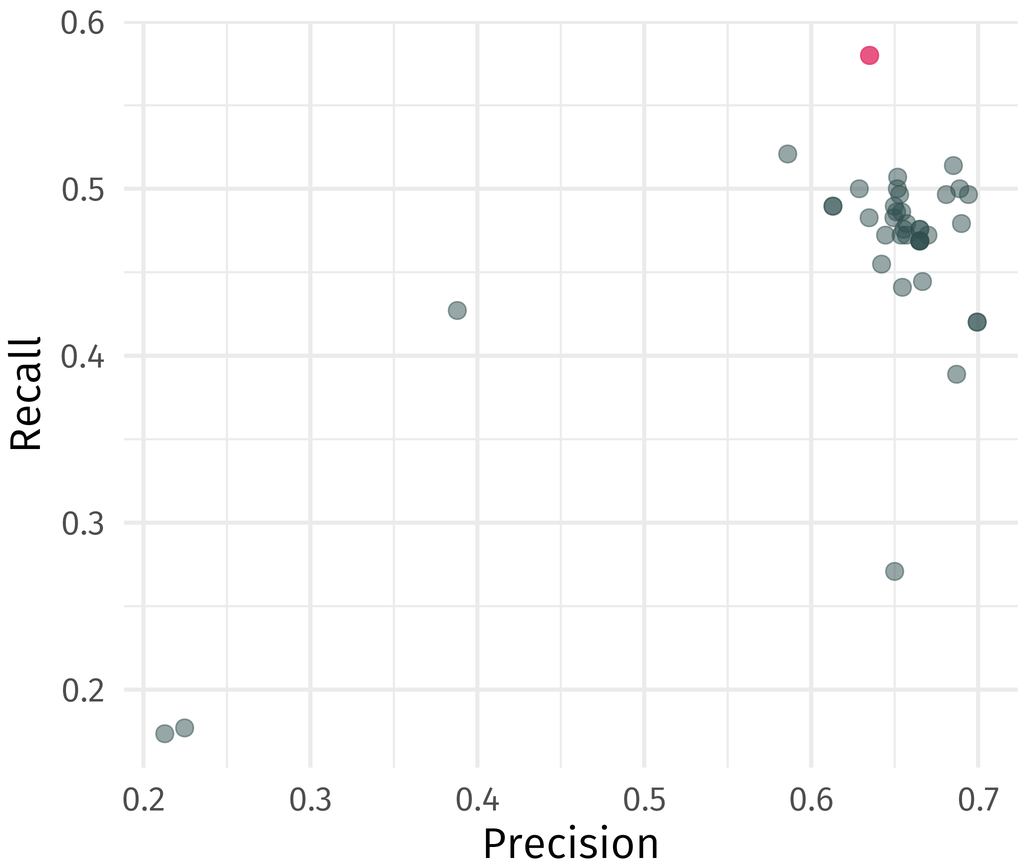
Results

In-class competition

Submission	Accuracy	Precision	Recall	F1
brad-bailey-simple-tree-model	0.791	0.665	0.469	0.550
coia_forest	0.789	0.657	0.472	0.549
coia_net	0.789	0.651	0.486	0.557
coia_tree	0.791	0.665	0.469	0.550
Craig_Submission	0.791	0.652	0.500	0.566
DNickles_cv_logistic_1_churn	0.802	0.689	0.500	0.579
DNickles_lasso_churn	0.793	0.699	0.420	0.525
DNickles_ridge_churn	0.793	0.699	0.420	0.525
Elliott_Eli_for	0.785	0.645	0.472	0.545
Elliott_Eli_net	0.789	0.650	0.490	0.558

In-class competition

Comparing (trading) precision and recall $\left(F_1 = 2 \times \dfrac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right)$



Support vector machines

Support vector machines

Intro

Support vector machines (SVMs) are a *general class* of classifiers that essentially attempt to separate two classes of observations.

> SVMs have been shown to perform well in a variety of settings, and are often considered one of the best "out of the box" classifiers. *ISL, p. 337*

The support vector machine generalizes a much simpler classifier—the maximal margin classifier.

The maximal margin classifier attempts to separate the **two classes** in our prediction space using **a single hyperplane**.

What's a hyperplane?

Consider a space with p dimensions.

A hyperplane is a $p - 1$ dimensional **subspace** that is

1. **flat** (no curvature)
2. **affine** (may or may not pass through the origin)

Examples

- In $p = 2$ dimensions, a *hyperplane* is a line.
- In $p = 3$ dimensions, a *hyperplane* is a plane.
- In $p = 1$ dimensions, a *hyperplane* is a point.

Support vector machines

Hyperplanes

We can define a **hyperplane** in p dimensions by constraining the linear combination of the p dimensions.[†]

For example, in two dimensions a hyperplane is defined by

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$

which is just the equation for a line.

Points $X = (X_1, X_2)$ that satisfy the equality *live* on the hyperplane.[††]

† Plus some offset ("intercept")

†† Alternatively: The hyperplane is composed of such points.

Support vector machines

Separating hyperplanes

More generally, in p dimensions, we defined a hyperplane by

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \tag{A}$$

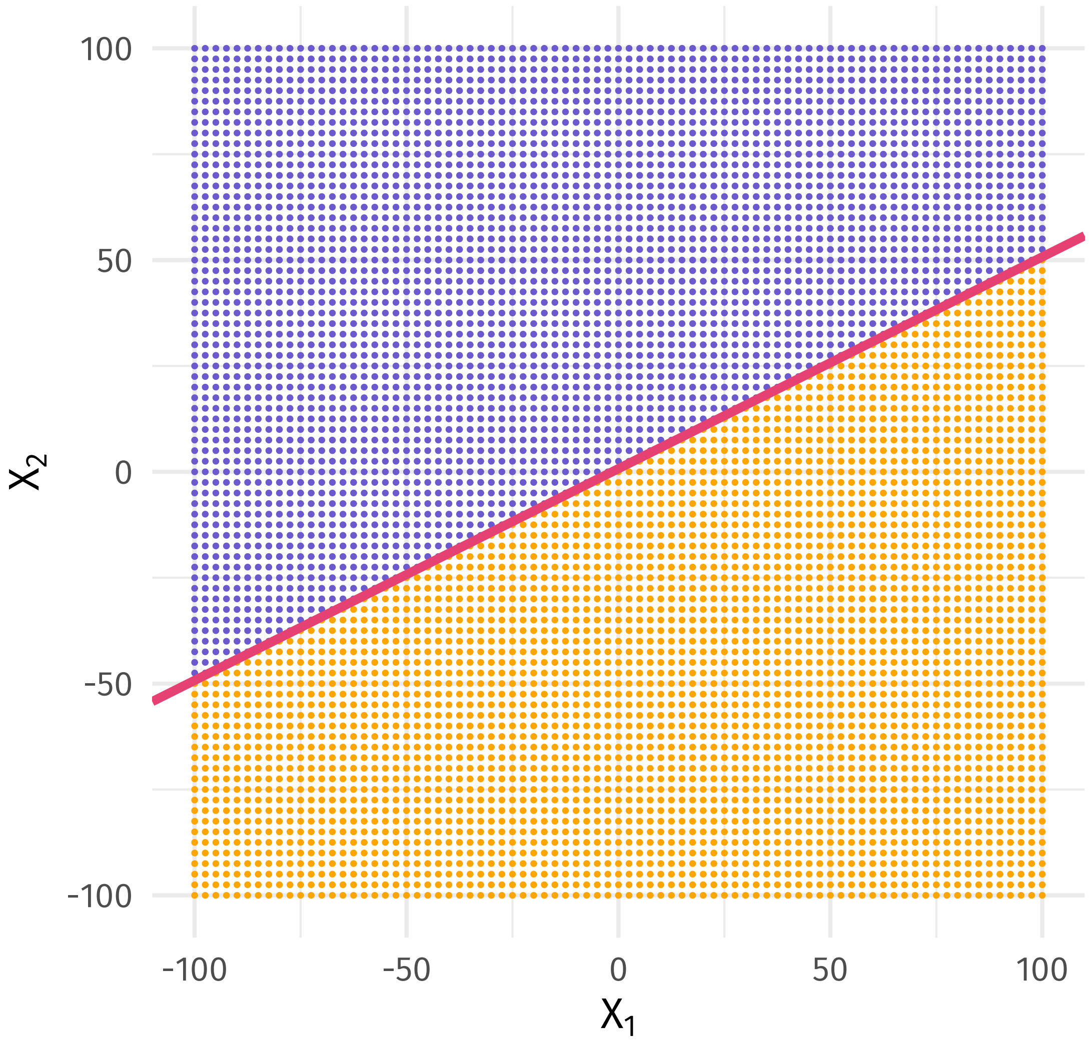
If $X = (X_1,\, X_2,\, \ldots,\, X_p)$ satisfies the equality, it is on the hyperplane.

Of course, not every point in the p dimensions will satisfy $\mathbf{A}$.

- If $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p > 0$, then X is ***above*** the hyperplane.
- If $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p - 0$, then X sits ***below*** the hyperplane.

The hyperplane *separates* the p-dimensional space into two "halves".

Ex: A **separating hyperplane** in two dimensions: $3 + 2X_1 - 4X_2 = 0$



Ex: A **separating hyperplane** in 3 dimensions: $3 + 2X_1 - 4X_2 + 2X_3 = 0$

trace 0

Support vector machines

Separating hyperplanes and classification

Idea: *Separate* two classes of outcomes in the p dimensions of our predictor space using a separating hyperplane.

To make a prediction for observation $(x^o, y^o) = (x_1^o, x_2^o, \ldots, x_p^o, y^o)$:

We classify points that live "above" of the plane as one class, *i.e.*,

$$\text{If } \beta_0 + \beta_1 x_1^o + \cdots + \beta_p x_p^o > 0\text{, then } \hat{y}^o = \text{Class 1}$$

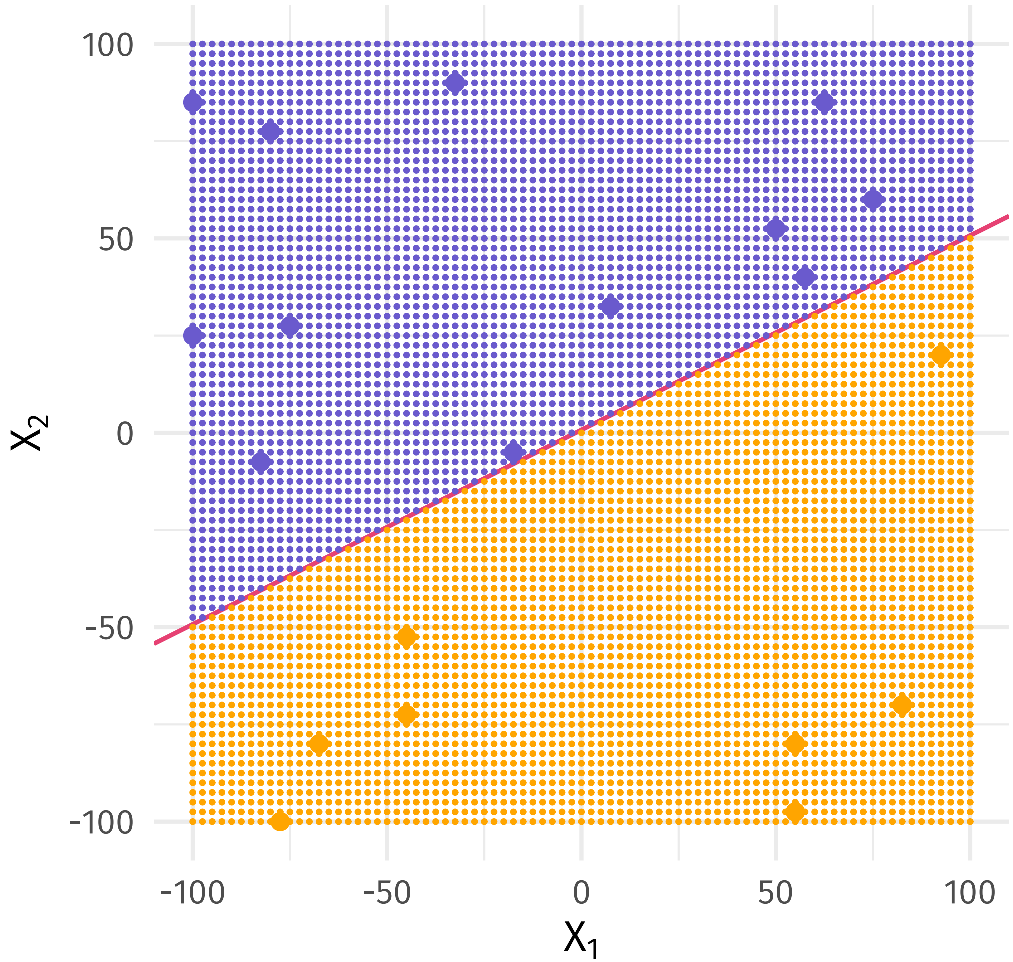
We classify points "below" the plane as the other class, *i.e.*,

$$\text{If } \beta_0 + \beta_1 x_1^o + \cdots + \beta_p x_p^o < 0\text{, then } \hat{y}^o = \text{Class 2}$$

Note This strategy assumes a separating hyperplane exists.

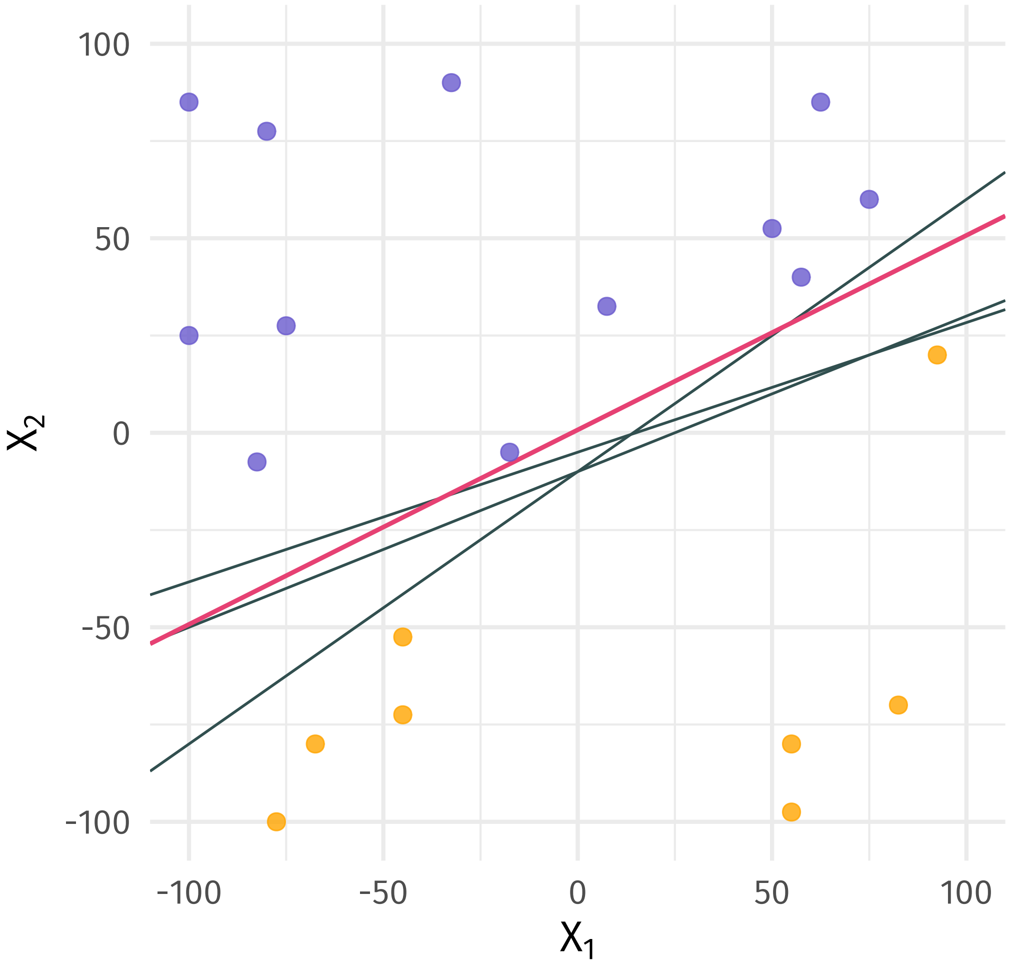
Support vector machines

If **a separating hyperplane** exists, then it defines a binary classifier.



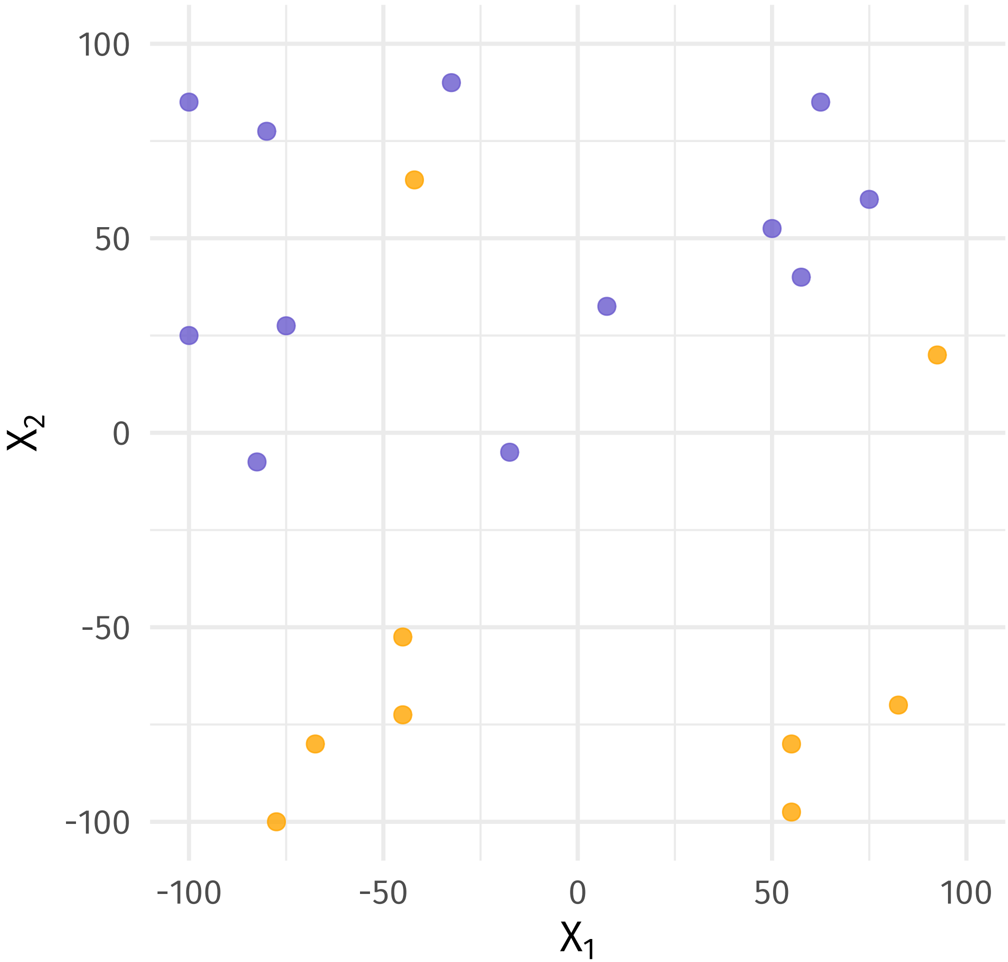
Support vector machines

If **a separating hyperplane** exists, then **many separating hyperplanes** exist.



Support vector machines

A **a separating hyperplane** may not exist.



Decisions

Summary A given hyperplane

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p = 0$$

produces a decision boundary.

We can determine any point's ($\boldsymbol{x}^o$) *side* of the boundary.

$$f(\boldsymbol{x}^o) = \beta_0 + \beta_1 x_1^o + \beta_2 x_2^o + \cdots + \beta_p x_p^o$$

We classify observationg $\boldsymbol{x}^o$ based upon whether $f(\boldsymbol{x}^o)$ is positive/negative.

The magnitude of $f(\boldsymbol{x}^o)$ tells us about our *confidence* in the classification.†

† Larger magnitudes are farther from the boundary.

Which separating hyperplane?

Q How do we choose between the possible hyperplanes?

A *One solution:* Choose the separating hyperplane that is "farthest" from the training data points—maximizing "separation."

The **maximal margin hyperplane**† is the hyperplane that

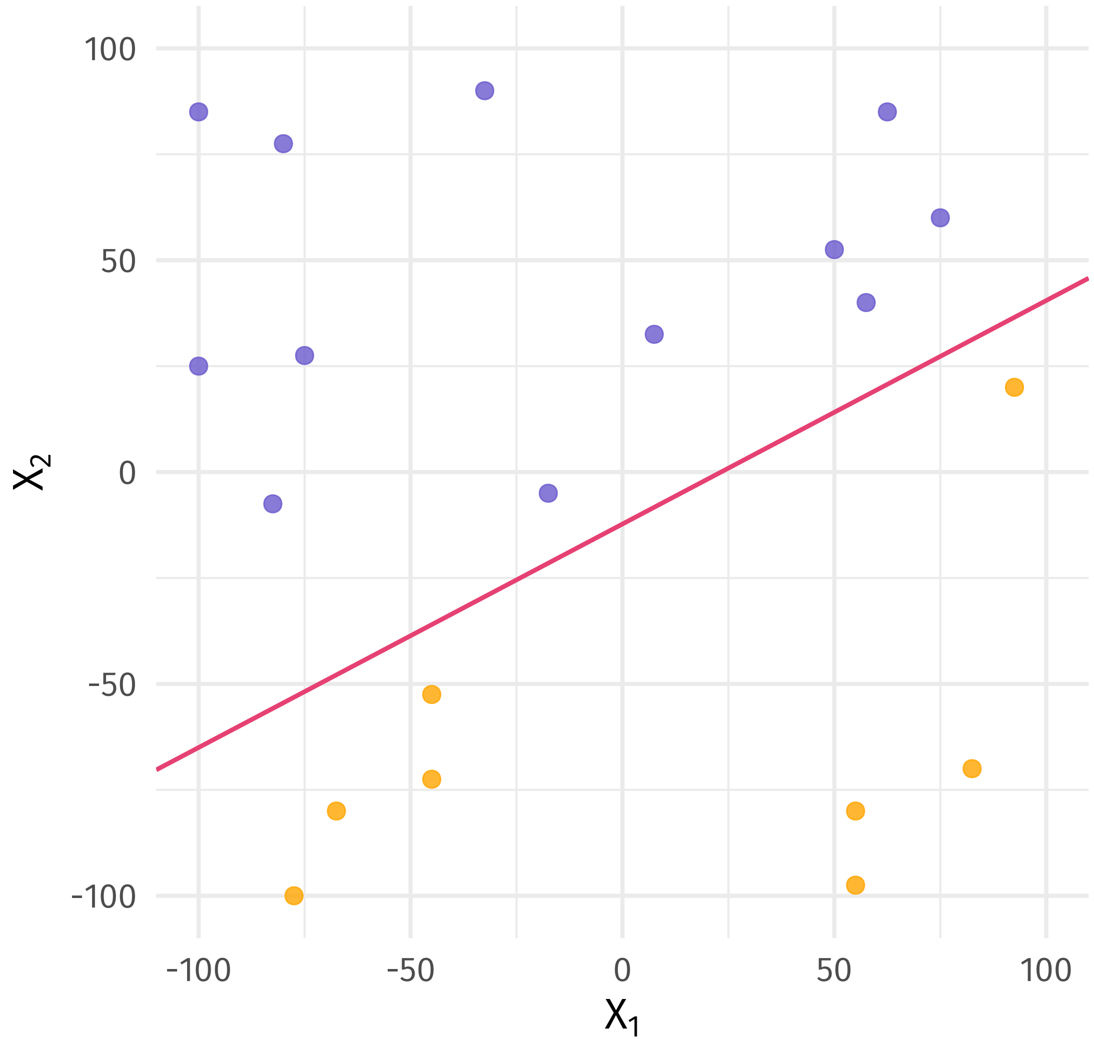
1. **separates** the two classes of obsevations
2. **maximizes** the **margin**—the distance to the nearest observation††

where *distance* is a point's perpendicular distance to the hyperplane.

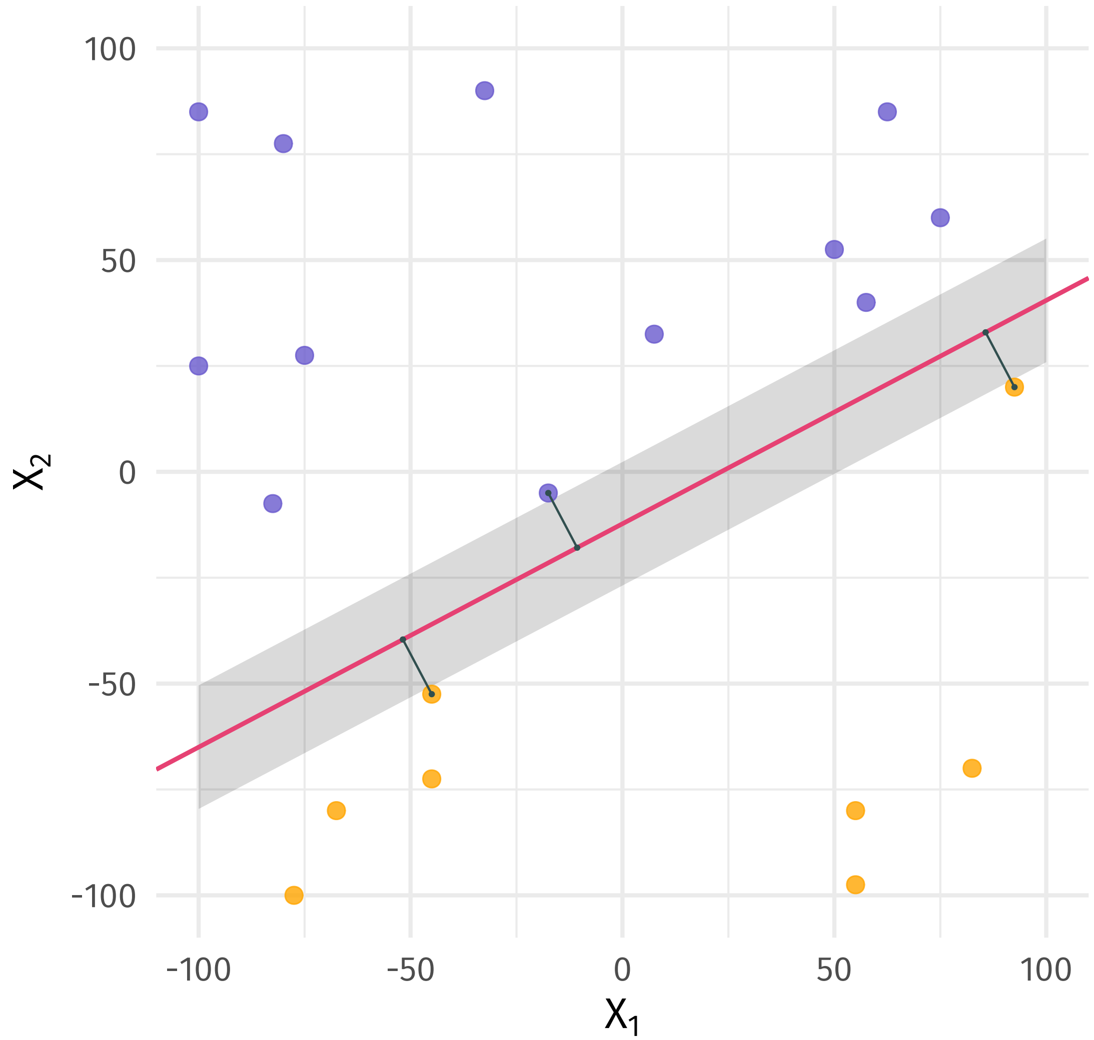
† AKA the *optimal separating hyperplane*

†† Put differently: The smallest distance to a training observation.

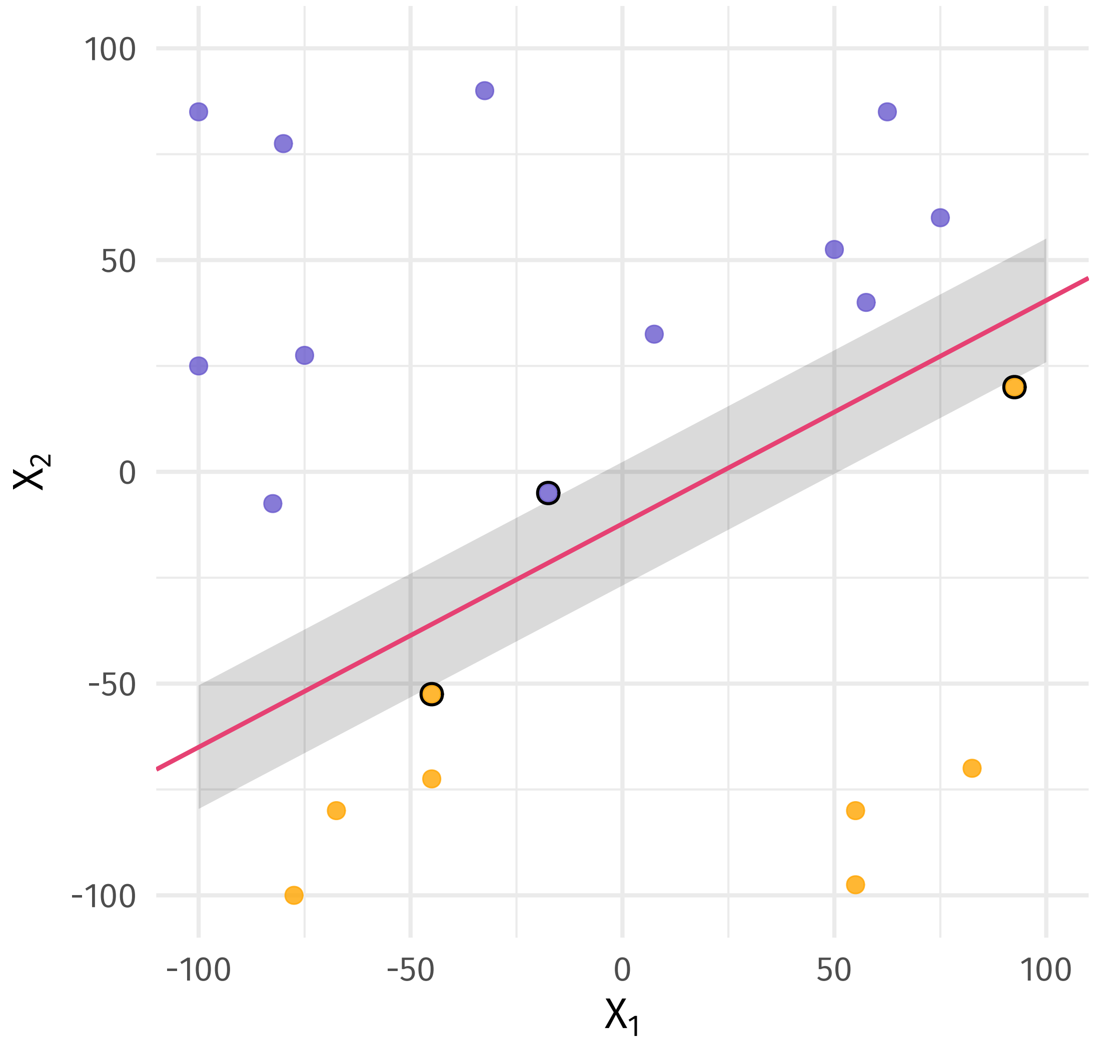
The **maximal margin hyperplane**...



…maximizes the **margin** between the hyperplane and training data…



…and is supported by three equidistant observations—the **support vectors**.



The maximal margin hyperplane

Formally, the maximal margin hyperplane solves the problem:

Maximize the margin M over the set of $\{\beta_0,\ \beta_1,\ \dots,\ \beta_p,\ M\}$ such that

$$\sum_{j=1}^{p} \beta_j^2 = 1 \tag{1}$$

$$y_i\left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}\right) \geq M \tag{2}$$

for all observations i.

(2) Ensures we separate (classify) observations correctly.

(1) allows us to interpret (2) as "distance from the hyperplane".

Fake constraints

Note that our first "constraint"

$$\sum_{j=1}^{p} \beta_j^2 = 1 \tag{1}$$

does not actually constrain $-1 \leq \beta_j \leq 1$ (or the hyperplane).

If we can define a hyperplane by

$$\beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_p x_{i,p} = 0$$

then we can also rescale the same hyperplane with some constant k

$$k\left(\beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_p x_{i,p}\right) = 0$$

The maximal margin classifier

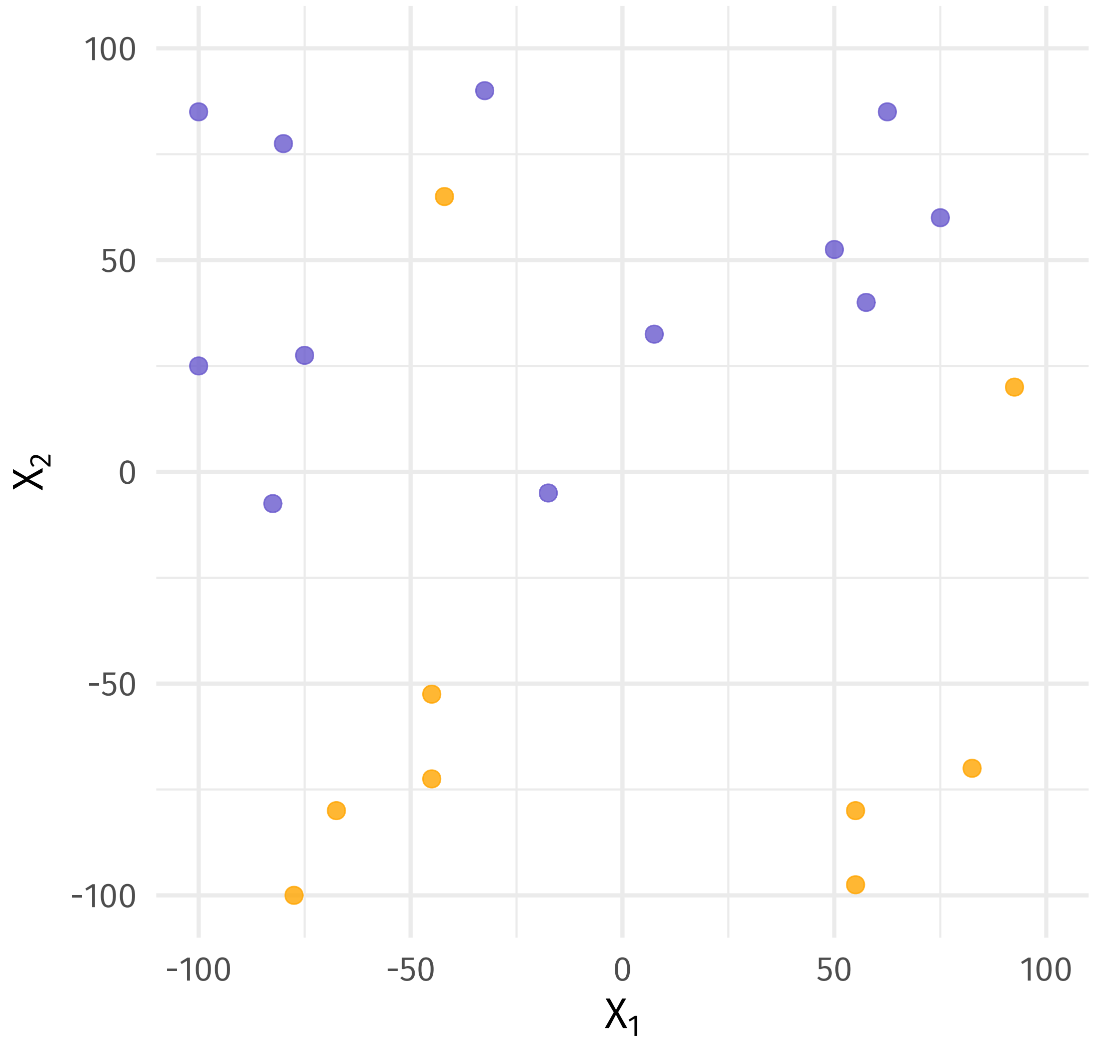
The maximal margin hyperplane produces the maximal margin classifier.

Notes

1. We are doing **binary classification**.
2. The decision boundary only uses the **support vectors**—very sensitive.
3. This classifier can struggle in **large dimensions** (big p).
4. A separating hyperplane does not always exist (**non-separable**).
5. Decision boundaries can be **nonlinear**.

Let's start by addressing non-separability...

Surely there's still a decent hyperplane-based classifier here, right?



Soft margins

When we cannot *perfectly* separate our classes, we can use **soft margins**, which are margins that "accept" some number of observations.

The idea: We will allow observations to be

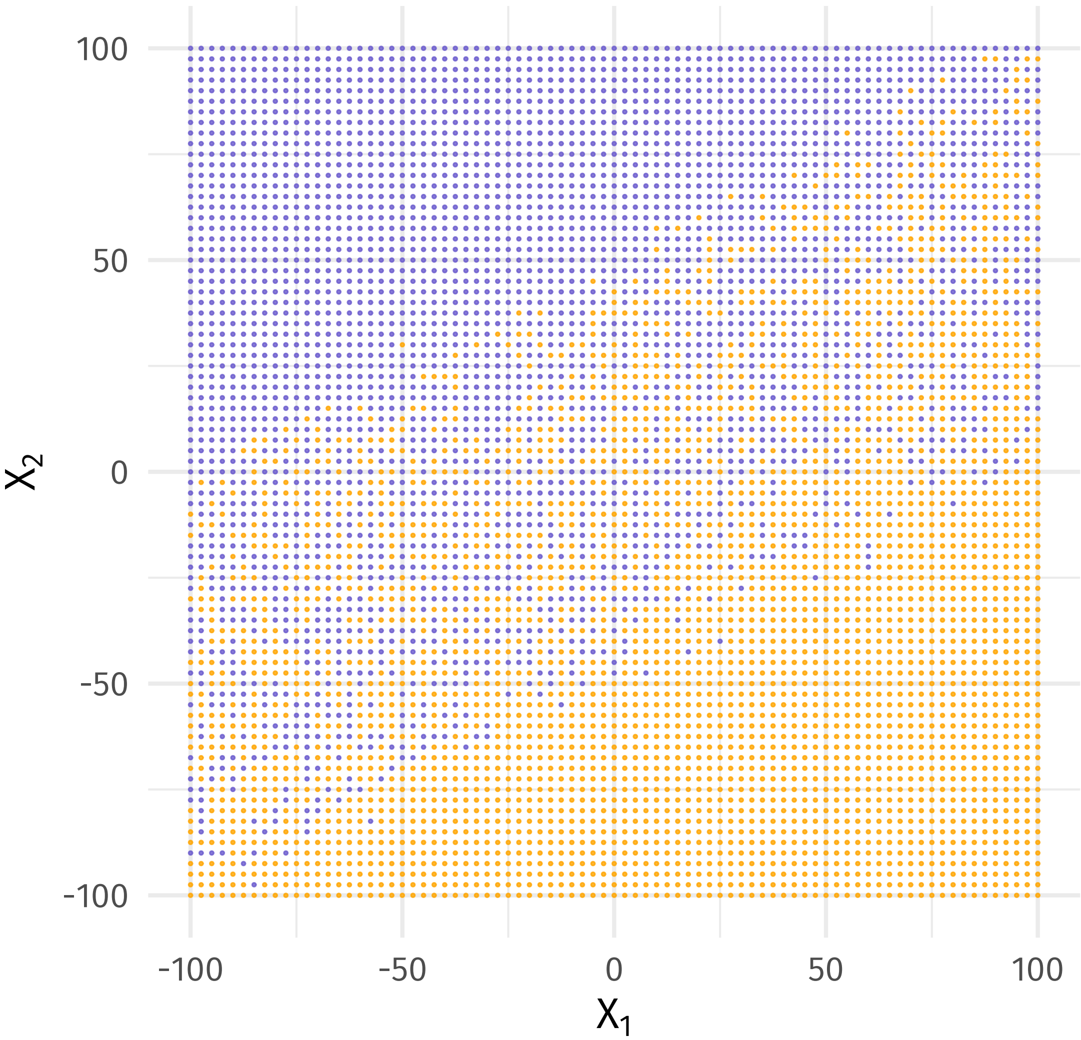
1. in the margin
2. on the wrong side of the hyperplane

but each will come with a price.

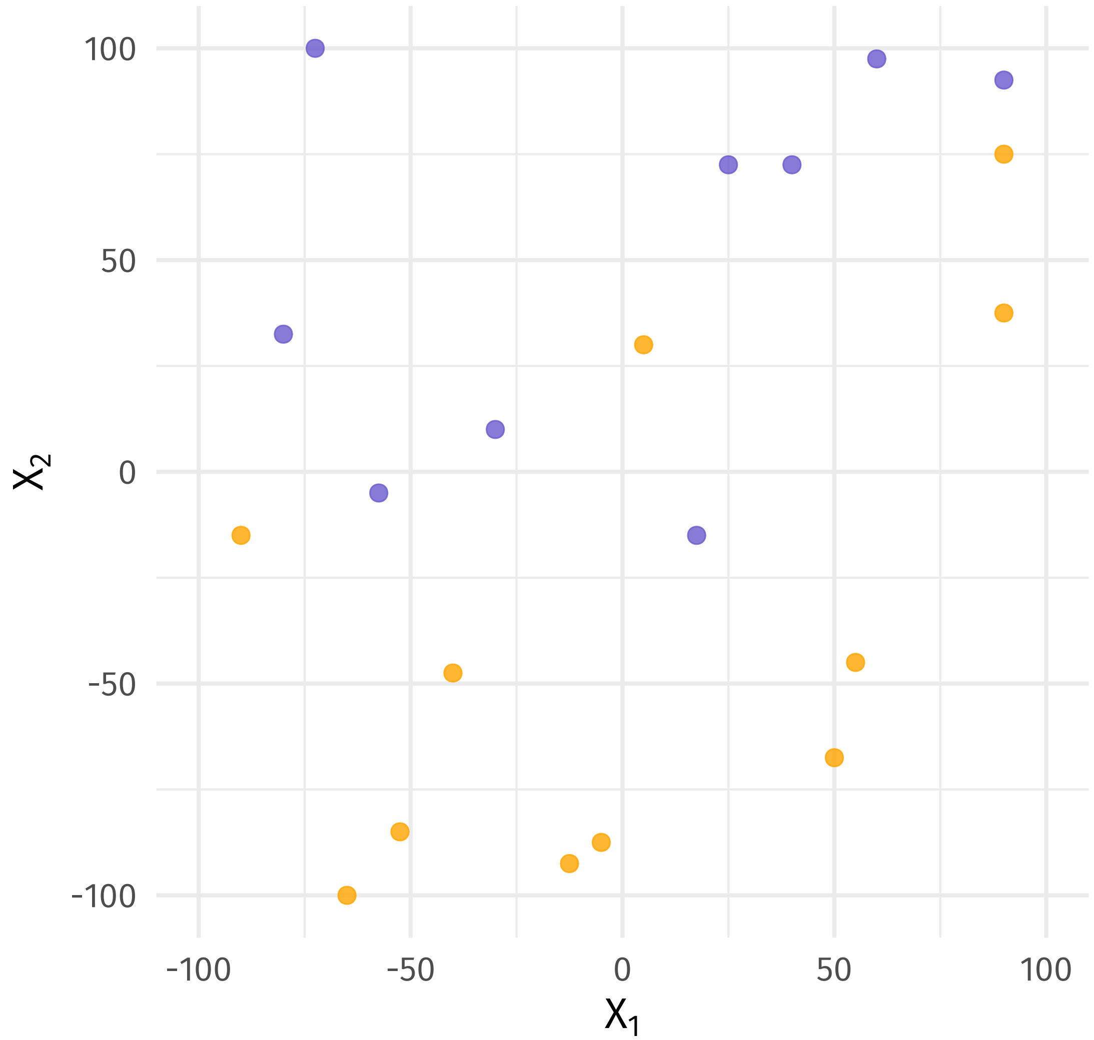
Using these *soft margins*, we create a hyperplane-based classifier called the **support vector classifier**.†

† Also called the *soft margin classifier*.

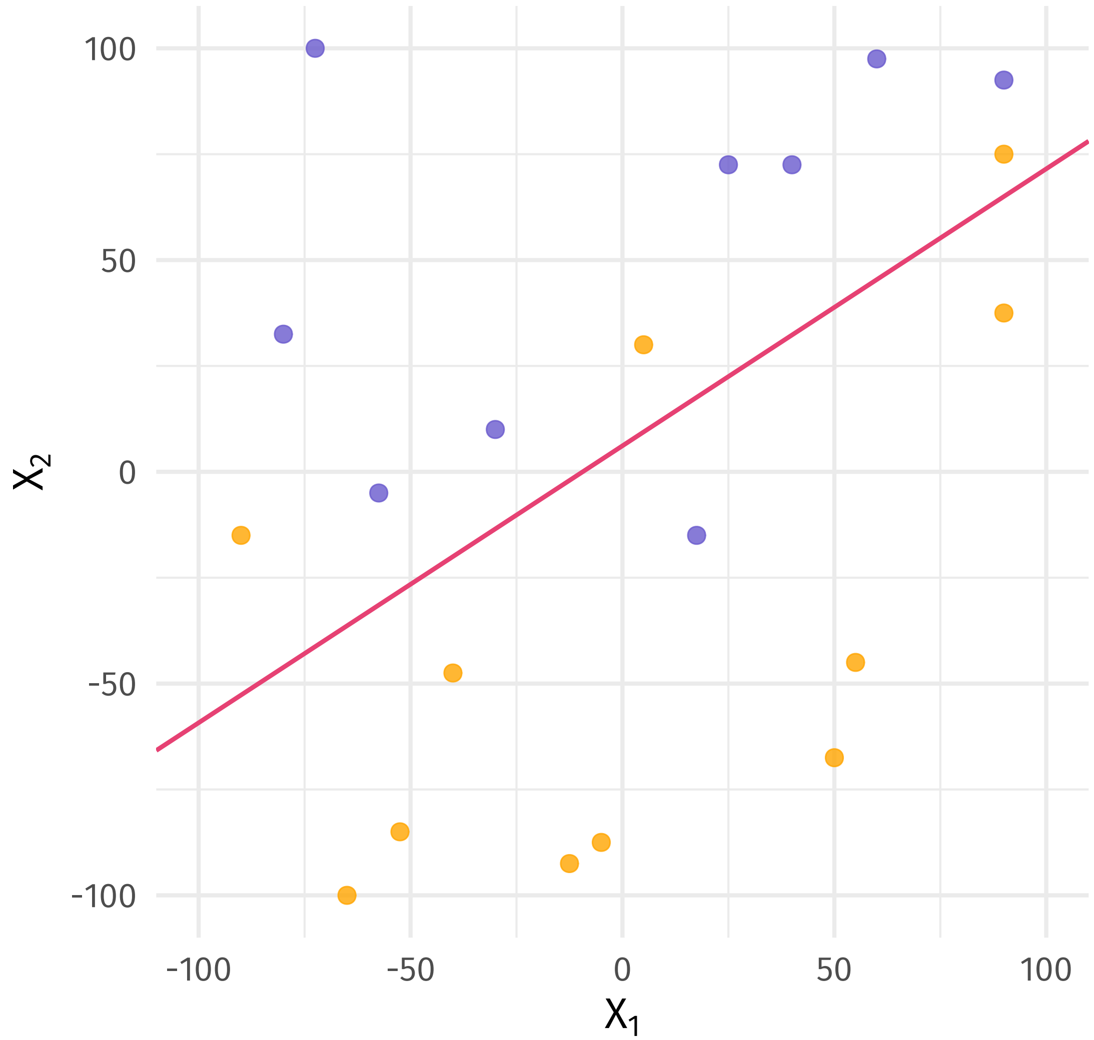
Our underlying population clearly does not have a separating hyperplane.



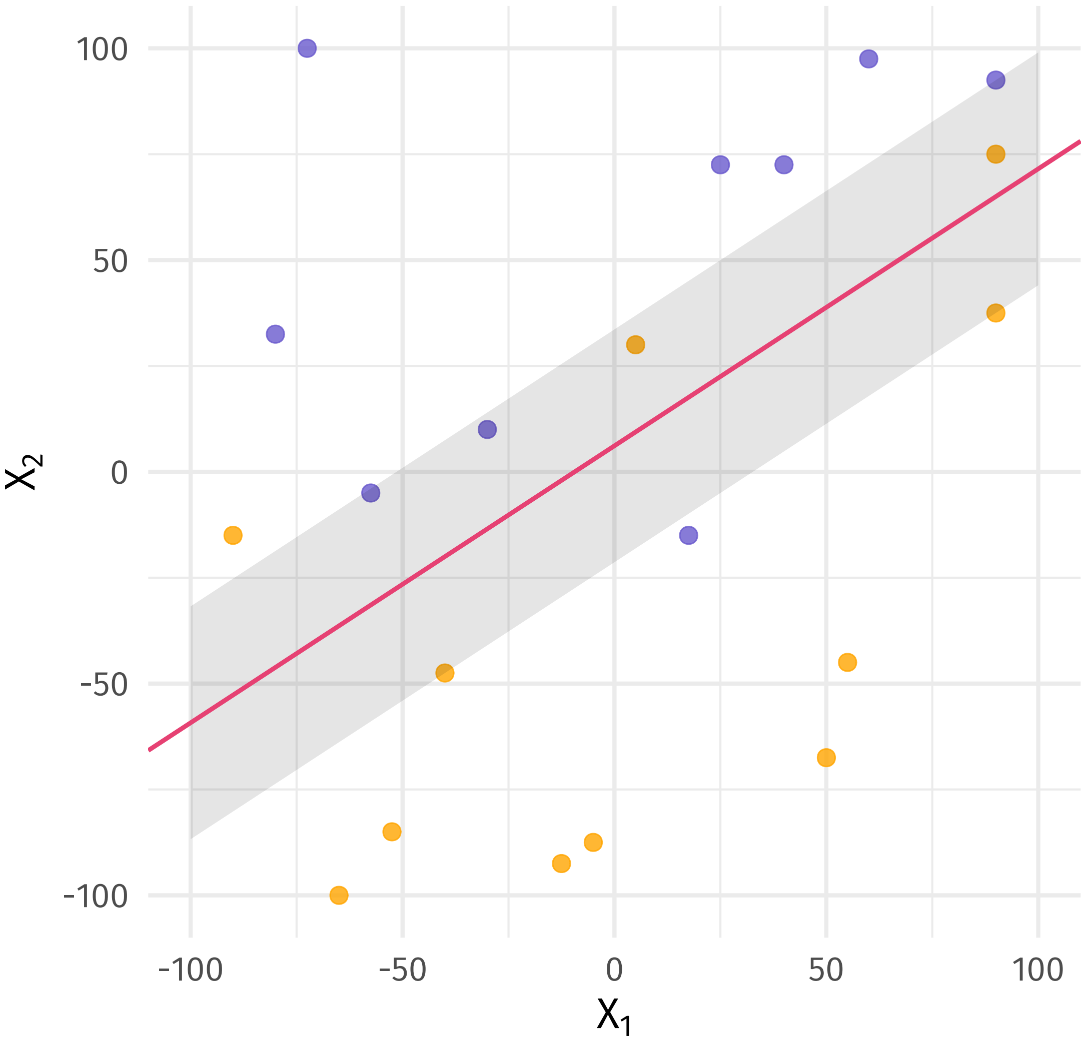
Our sample population also does not have a separating hyperplane.



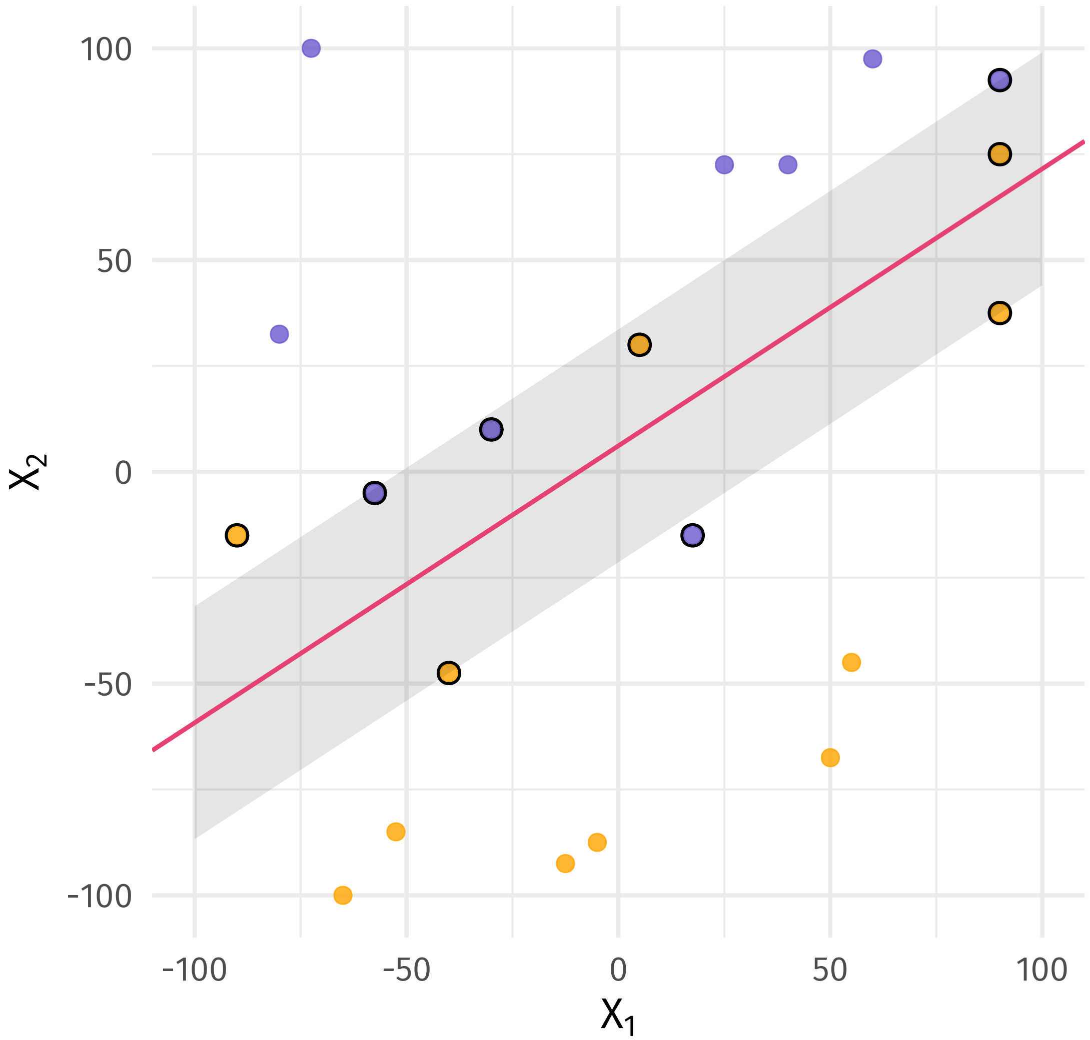
Our **hyperplane**



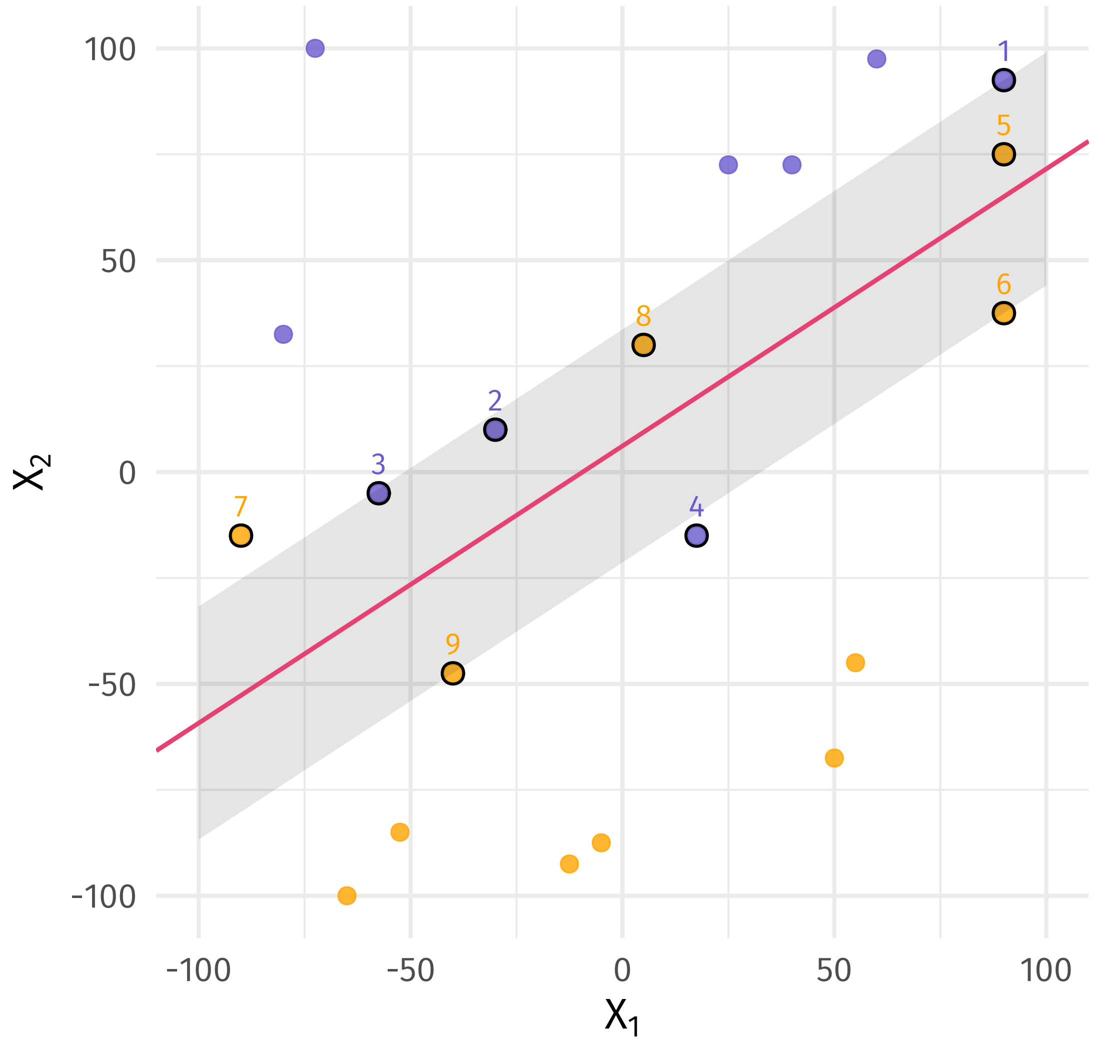
Our **hyperplane** with **soft margins**...



Our **hyperplane** with **soft margins** and **support vectors**.



Support vectors: on (i) the margin or (ii) on the wrong side of the margin.



Support vector classifier

The support vector classifier selects a hyperplane by solving the problem

Maximize the margin M over the set $\{\beta_0, \beta_1, \ldots, \beta_p, \epsilon_1, \ldots, \epsilon_n, M\}$ s.t.

$$\sum_{j=1}^{p} \beta_j^2 = 1 \tag{3}$$

$$y_i\left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}\right) \geq M\left(1 - \epsilon_i\right) \tag{4}$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C \tag{5}$$

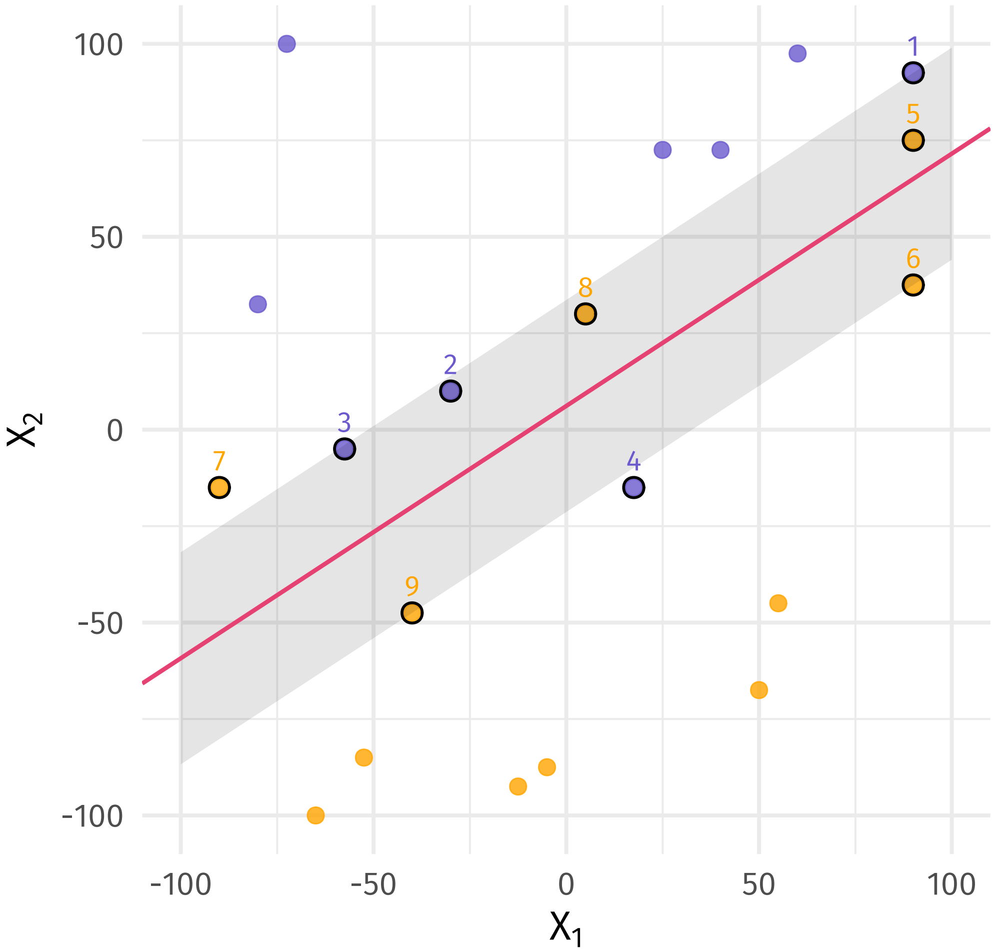
The ϵ_i are slack variables that allow i to *violate* the margin or hyperplane.
C gives is our budget for these violations.

Let's consider constraints (4) and (5) work together...

$$y_i\left(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_p x_{ip}\right) \geq M\left(1-\epsilon_i\right) \tag{4}$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^n \epsilon_i \leq C \tag{5}$$

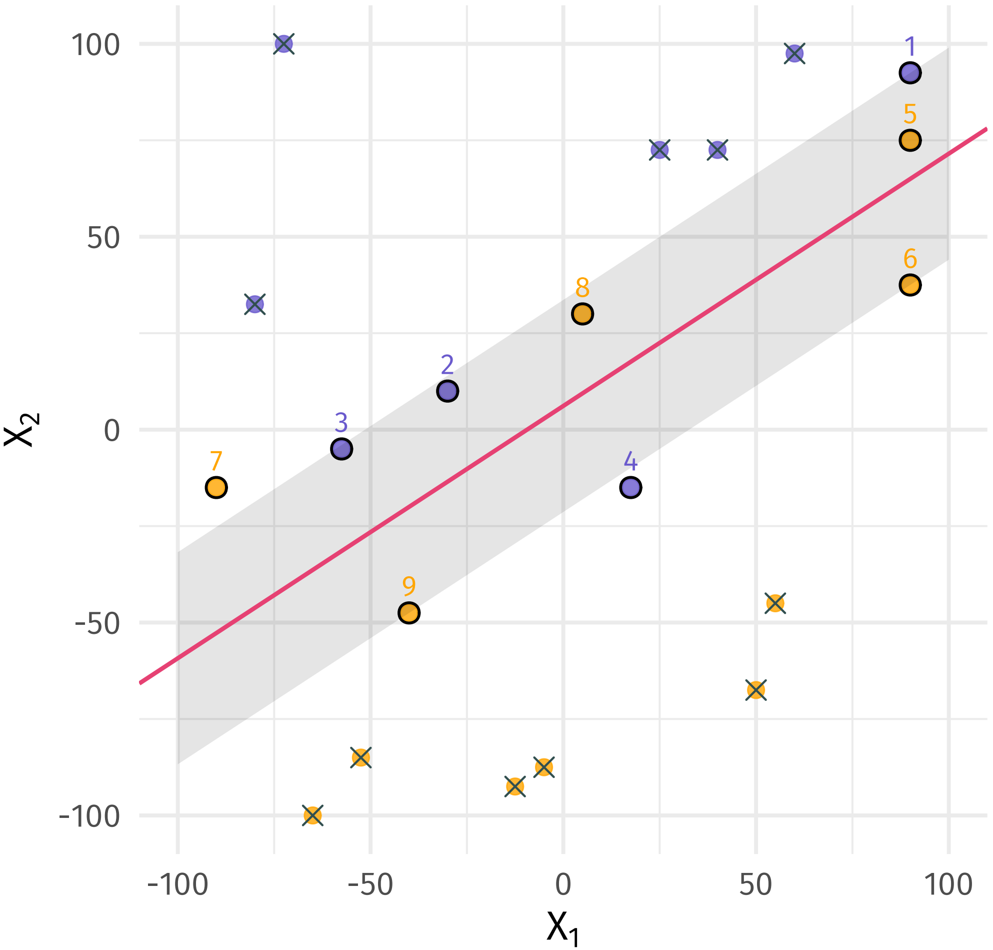
$$y_i\left(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_p x_{ip}\right) \geq M\left(1-\epsilon_i\right), \quad \epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C$$



For $\epsilon_i = 0$:

- $M\left(1-\epsilon_i\right) > 0$
- Correct side of hyperplane
- Correct side of margin (or on margin)
- No cost (C)
- Distance $\geq M$
- *Examples?*

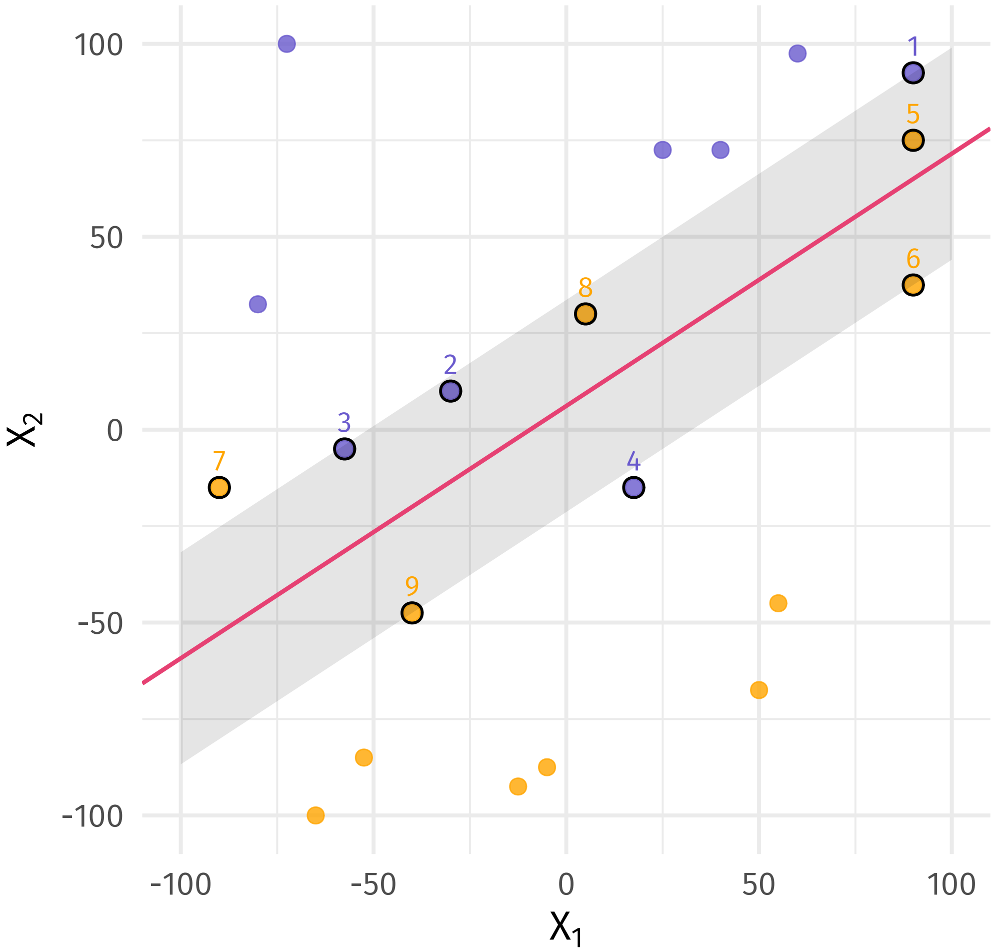
$$y_i\left(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_p x_{ip}\right) \geq M\left(1-\epsilon_i\right), \quad \epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C$$



For $\epsilon_i = 0$:

- $M\left(1-\epsilon_i\right) > 0$
- Correct side of hyperplane
- Correct side of margin (or on margin)
- No cost (C)
- Distance $\geq M$
- Correct side of margin: $(\times)$
- *On margin:* 1, 6, 9

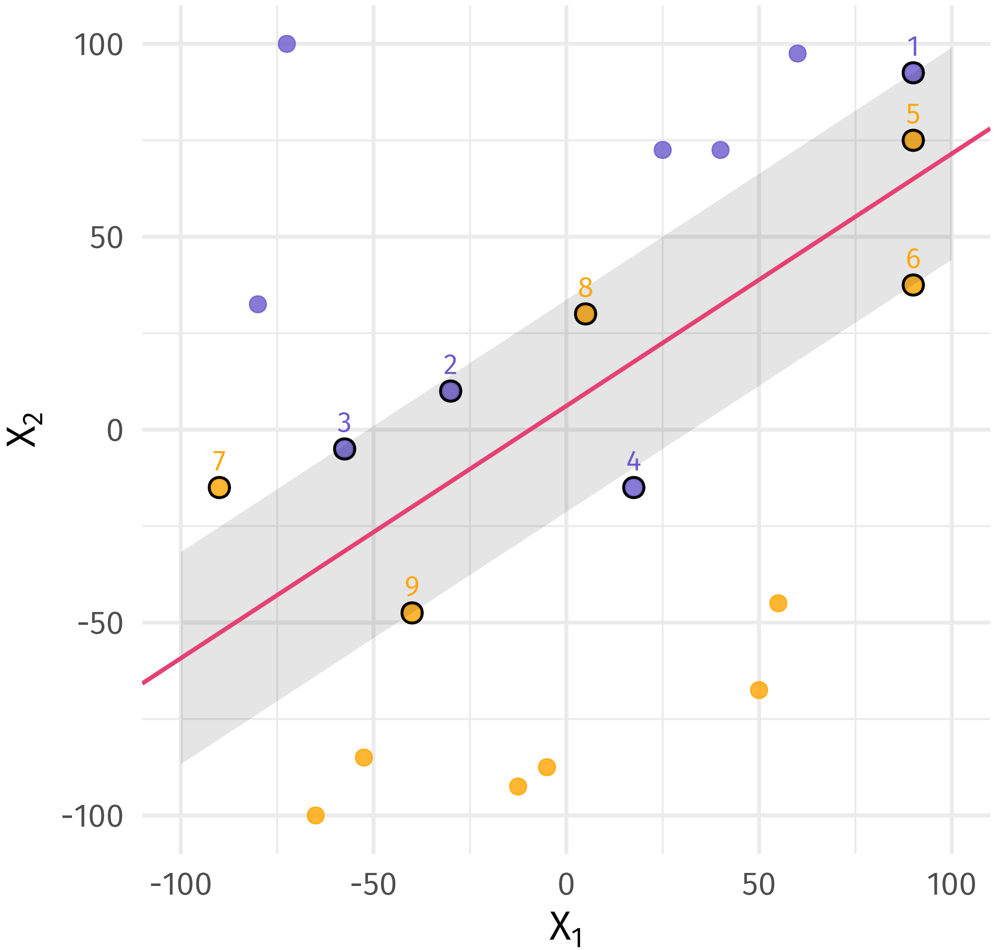
$$y_i\left(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_p x_{ip}\right) \geq M\left(1-\epsilon_i\right), \quad \epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C$$



For $0 \leq \epsilon_i \leq 1$:

- $M\left(1-\epsilon_i\right) > 0$
- Correct side of hyperplane
- Wrong side of the margin (*violates margin*)
- Pays cost ϵ_i
- Distance $< M$
- *Examples?*

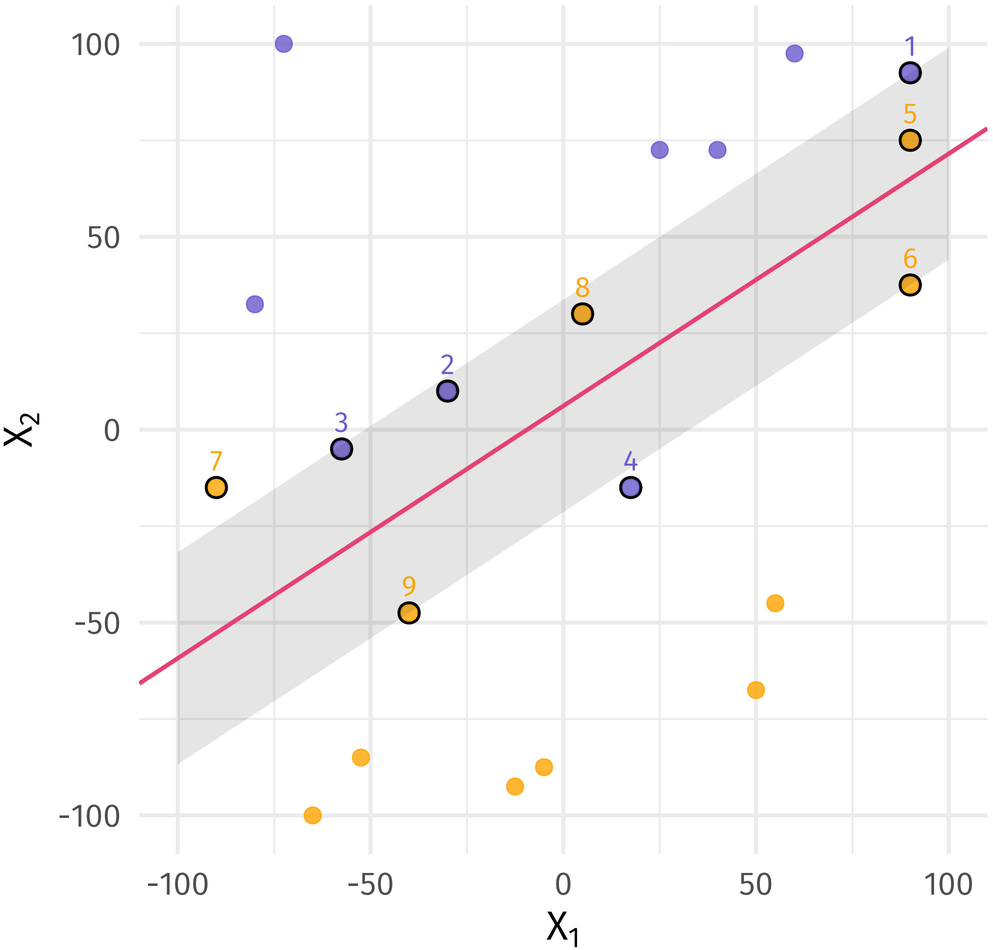
$$y_i\left(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_p x_{ip}\right) \geq M\left(1-\epsilon_i\right), \quad \epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C$$



For $0 \leq \epsilon_i \leq 1$:

- $M\left(1-\epsilon_i\right)>0$
- Correct side of hyperplane
- Wrong side of the margin (*violates margin*)
- Pays cost ϵ_i
- Distance $< M$
- *Ex:* 2, 3

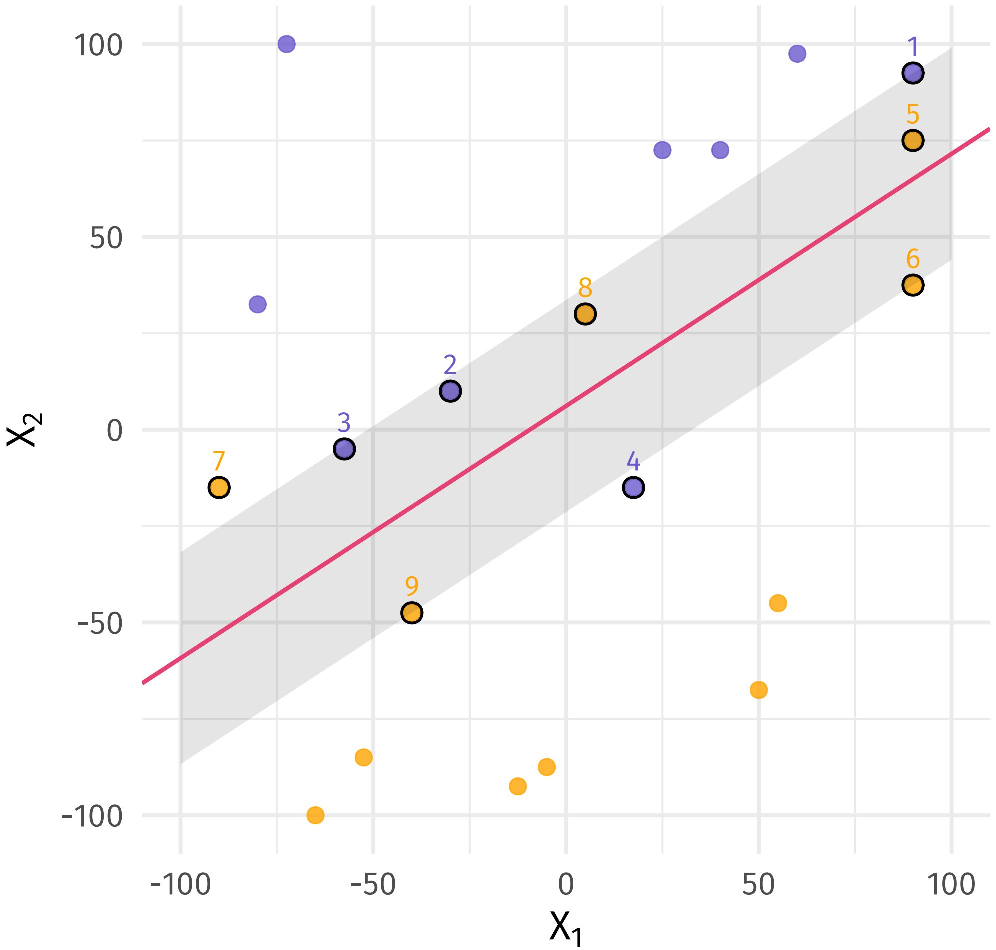
$$y_i\left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}\right) \geq M\left(1 - \epsilon_i\right), \quad \epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C$$



For $\epsilon_i \geq 1$:

- $M\left(1 - \epsilon_i\right) < 0$
- Wrong side of hyperplane
- Pays cost ϵ_i
- Distance $\lesseqqgtr M$
- *Examples?*

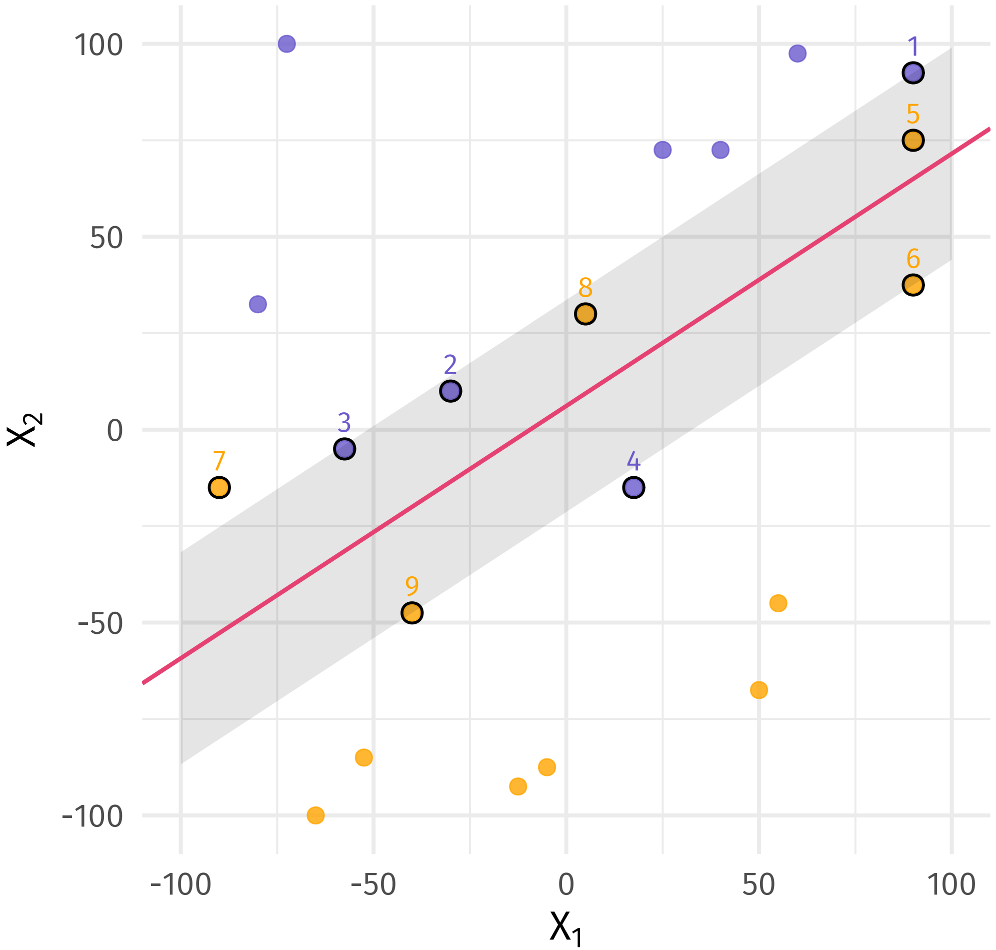
$$y_i\left(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_p x_{ip}\right) \geq M\left(1-\epsilon_i\right), \quad \epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C$$



For $\epsilon_i \geq 1$:

- $M\left(1-\epsilon_i\right)<0$
- Wrong side of hyperplane
- Pays cost ϵ_i
- Distance $\lesseqqgtr M$
- *Ex:* 4, 5, 7, 8

$$y_i\left(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_p x_{ip}\right) \geq M\left(1-\epsilon_i\right), \quad \epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C$$



Support vectors

- On margin
- Violate margin
- Wrong side of hyperplane

determine the classifier.

Support vector classifier

The tuning parameter C determines how much *slack* we allow.

C is our budget for violating the margin—including observations on the wrong side of the hyperplane.

Case 1: $C = 0$

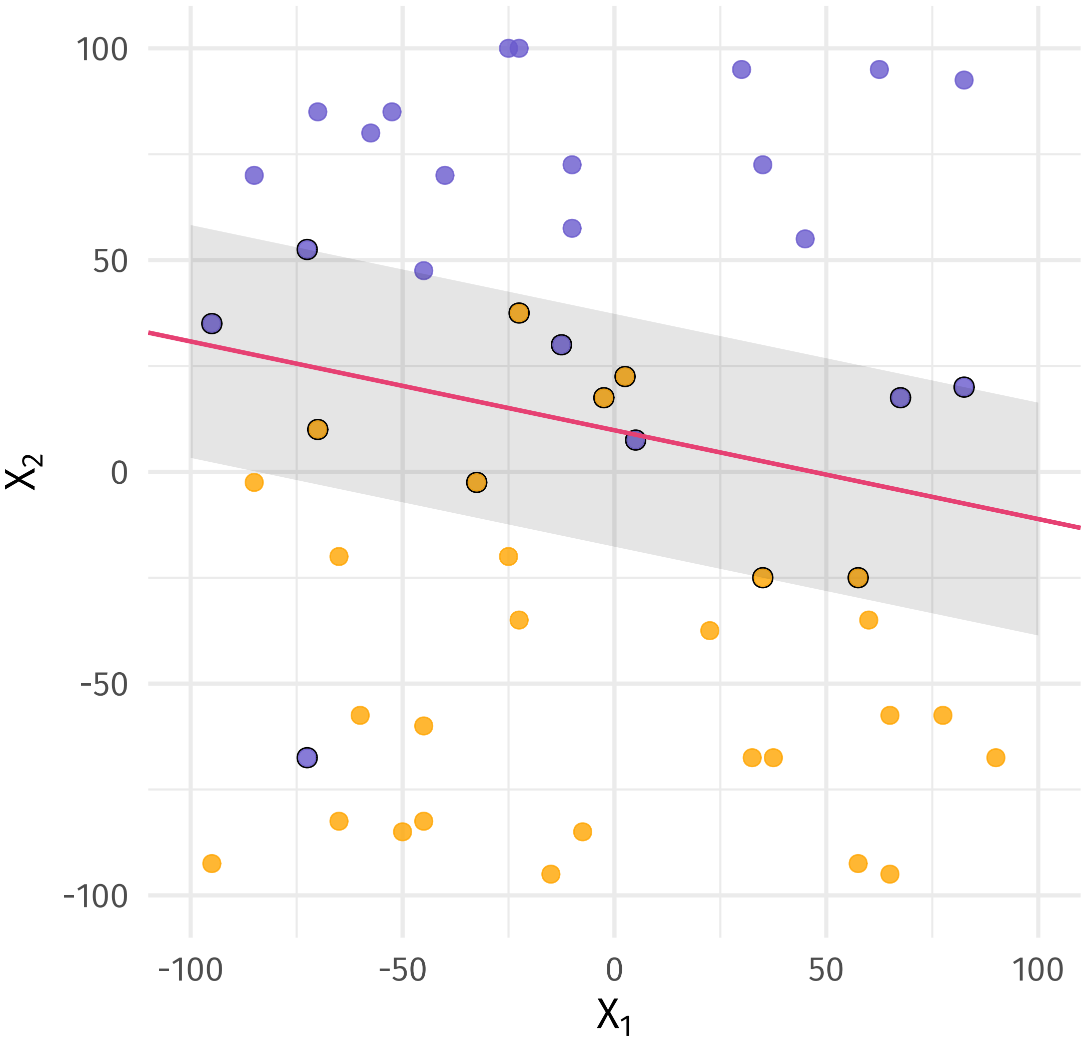
- We allow no violations.
- Maximal margin hyperplane.
- Trains on few obs.

Case 2: $C > 0$

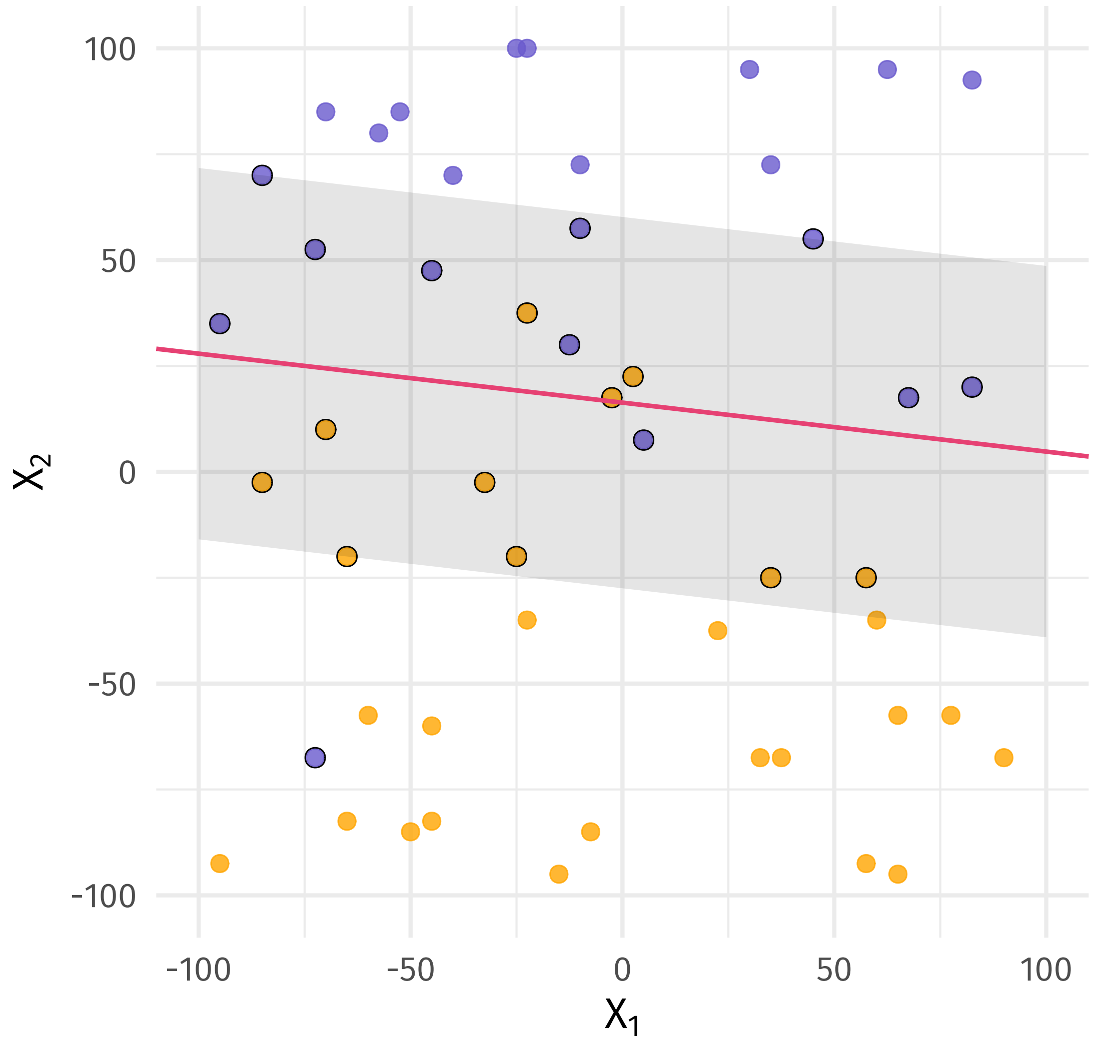
- $\leq C$ violations of hyperplane.
- *Softens* margins
- Larger C uses more obs.

We tune C via CV to balance low bias (low C) and low variance (high C).

Starting with a low budget (C).



Now for a high budget (C).



Sources

These notes draw upon

- An Introduction to Statistical Learning (*ISL*)
  James, Witten, Hastie, and Tibshirani

Table of contents

Admin

SVM

Other