LDA, QDA & KNN

AU STAT-427/627

Emil Hvitfeldt

2021-5-26

Classification

We have 2 or more groups in our data and we want to create rules to detect/classify them

We looked at logistic regression last week

This week we will explore 3 more methods

- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)
- K-Nearest Neighbor (KNN)

LDA

logistic regression tries to modeling $Pr(Y = k|X = x)$ directly

Model the distribution of predictors X in each of the classes k. Then use Bayes' theorem to flip these around to create estimates for $Pr(Y = k|X = x)$

LDA

- LDA is more stable than logistic regression when the classes are well separated

- If n is small and the distribution of the predictors are approximately normal in each of the classes LDA is more stable than logistic regression

- LDA naturally extends to work with more than 2 classes

LDA

For simplicity, we start with the case of 1 predictor

Notation:

$$f_k(x) = Pr(X = x|Y = k)$$

denote the **density function** of X for an observation that comes from the kth class.

- $f_k(x)$ is large if there is a high probability that an observation is part of class k when $X = k$

- $f_k(x)$ is small if there is a low probability that an observation is part of class k when $X = k$

Bayes' Theorem

let π_k represent the overall(prior) probability that a randomly chosen observation is associated with the kth class

We have that

$$Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

with the abbreviation $p_k(X) = Pr(Y = k|X)$

Bayes' Theorem

$$Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

Idea:

estimate π_k and $f_k(x)$ and plug those in instead of directly computing $p_k(X)$

LDA

To estimate $f_k(x)$ we start by making some assumptions for $f_k(x)$

- We assume that $f_k(x)$ is **normal**, for 1-dimensional the normal density takes the form

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)$$

where μ_k and σ_k^2 are the mean and variance for the kth class

LDA

To estimate $f_k(x)$ we start by making some assumptions for $f_k(x)$

- We assume that $\sigma_1^2 = \ldots = \sigma_K^2$: there is a shared variance term across all K classes

We denote this shared variable by σ^2

Plugging everything in we get

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}$$

LDA

Taking the log and rearranging some terms we an equivalent function

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

which we what to find the largest value for

So now we just have to estimate the means $\mu_1, \ldots, \mu_K$, prior probabilities $\pi_1, \ldots, \pi_K$ and the shared variance σ^2

So we have $2K + 1$ parameters to estimate

LDA

The estimate we will use are

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma} = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

where n is the number of observations overall and n_k is the number of observations in the Kth class

LDA

We also estimate the prior probabilities with

$$\hat{\pi}_k = \frac{n_k}{n}$$

We plug in our estimate and get the estimator

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

LDA

We need to extend the LDA classifier to work with multiple predictors

For this, to work we assume that $X = (X_1, X_2, \ldots, X_p)$ is drawn from a multivariate normal distribution, with a class-specific mean vector and a common covariance matrix.

LDA

The multivariate normal density is defined as

$$f(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

LDA

Plugging in a rearranging we get

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^Y \Sigma^{-1} \mu_k + \log(\pi_k)$$

This is the vector/matrix version of what we saw for $p = 1$

if $p = 1$ then this simplifies back to the earlier case

LDA

the linear discriminant analysis gets its name because the discriminant function is a linear combination of x and the decision boundary is linear

QDA

We made a couple of assumptions of the distribution of the predictors X to construct the LDA classifier

If we relax the assumption that each class has its own covariance matrix then we get the quadratic discriminant analysis (QDA) model

We assume that an observation from the kth class is of the form $X \sim N(\mu_k, \Sigma_k)$ where Σ_k is the covariance matrix for the kth class.

QDA

The discriminant function under this assumption has the form

$$\delta_k(x) = -\frac{1}{2}x^T\Sigma_k^{-1}x + x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^Y\Sigma^{-1}\mu_k - \frac{1}{2}\log|\Sigma_k| + \log(\pi_k)$$

with these terms being new for the QDA over LDA

QDA gets its name because the discriminant function is quadratic in x

We end up getting a quadratic decision boundaries

Why would one prefer QDA over LDA?

Bias-variance trade-off!

LDA has a lot fewer parameters then QDA

$$\frac{p(p+1)}{2} \quad vs \quad \frac{Kp(p+1)}{2}$$

LDA is a much less flexible classifier (partly because it is linear) and has a lower variance

Why would one prefer QDA over LDA?

If LDA's assumption that the K classes share a common covariance matrix is badly off, then LDA can suffer from high bias.

If you have few observation and you want to reduce variance then you need to use LDA other QDA

In the end, if you have a linear decision boundary in your data then an LDA will work just as good as a QDA but the QDA will have a higher variance since it needs to estimate a larger number of parameters

LDA or logistic regression

Only applicable for $K = 2$

Logistic regression

The groups may have quite different n

Not so sensitive to outliers

concentrates more on examples near the class boundary and basically disregards cases at the "backside" of the distributions.

LDA

The groups should have similar n

Quite sensitive to outliers

KNN

k-nearest neighbor was introduced in the second chapter and we will catch up this week

We want to estimate the conditional distribution of Y given X and classify an observation to the class with the highest probability

K-nearest neighbor takes this literally and classifies an observation solely based on what the classes of its neighbors would be in the training data set.

KNN

$$Pr(Y = k|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = k)$$

For $K = 1$ the algorithm predicts the new points only according to the closest neighbors

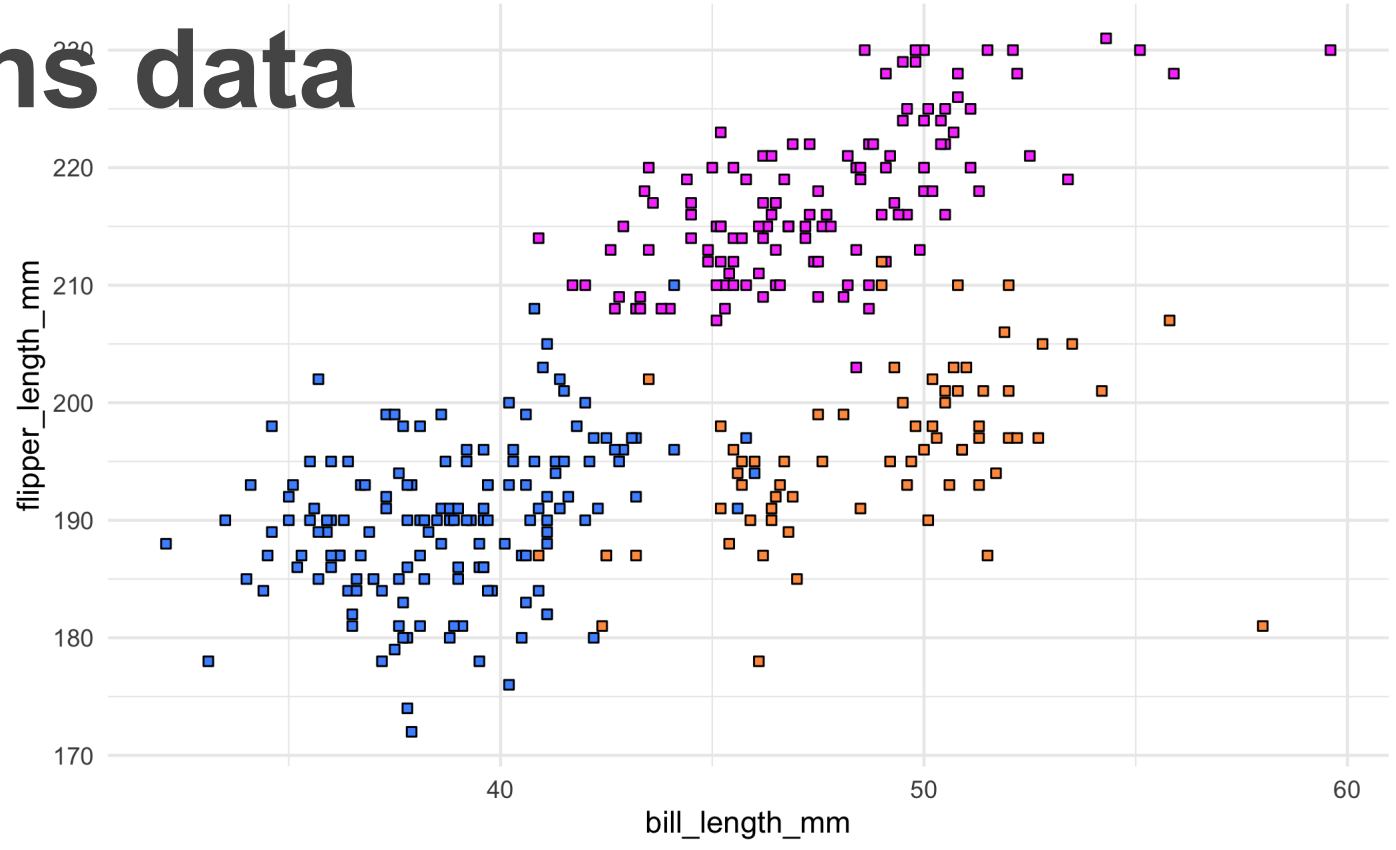
For $k = 5$ the algorithm predicts whichever class appears most often

KNN

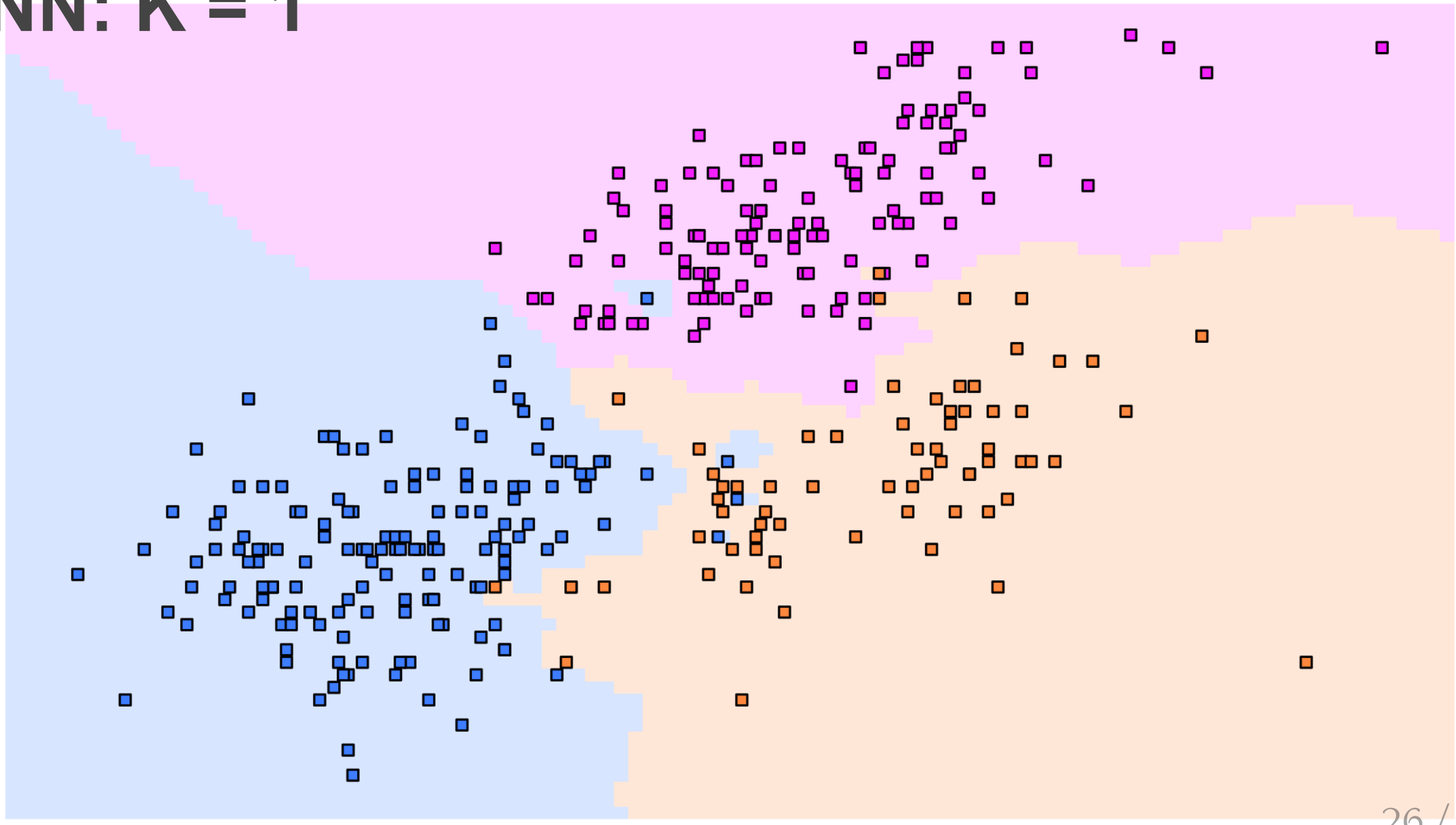
K is typically taken to be odd to avoid ties

We don't really do any modeling, the model queries the training data to find the neighbors for new points

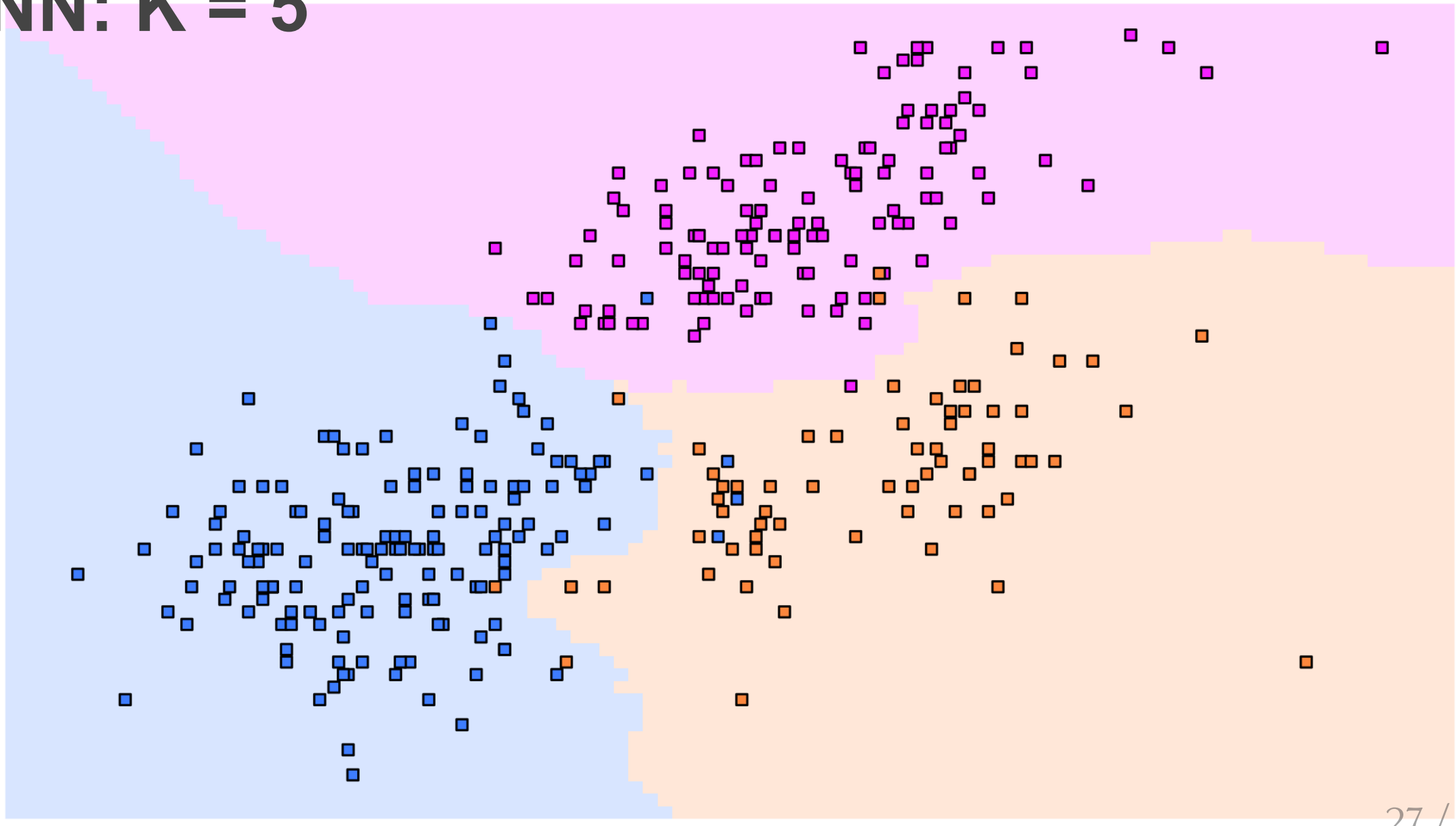
Penguins data



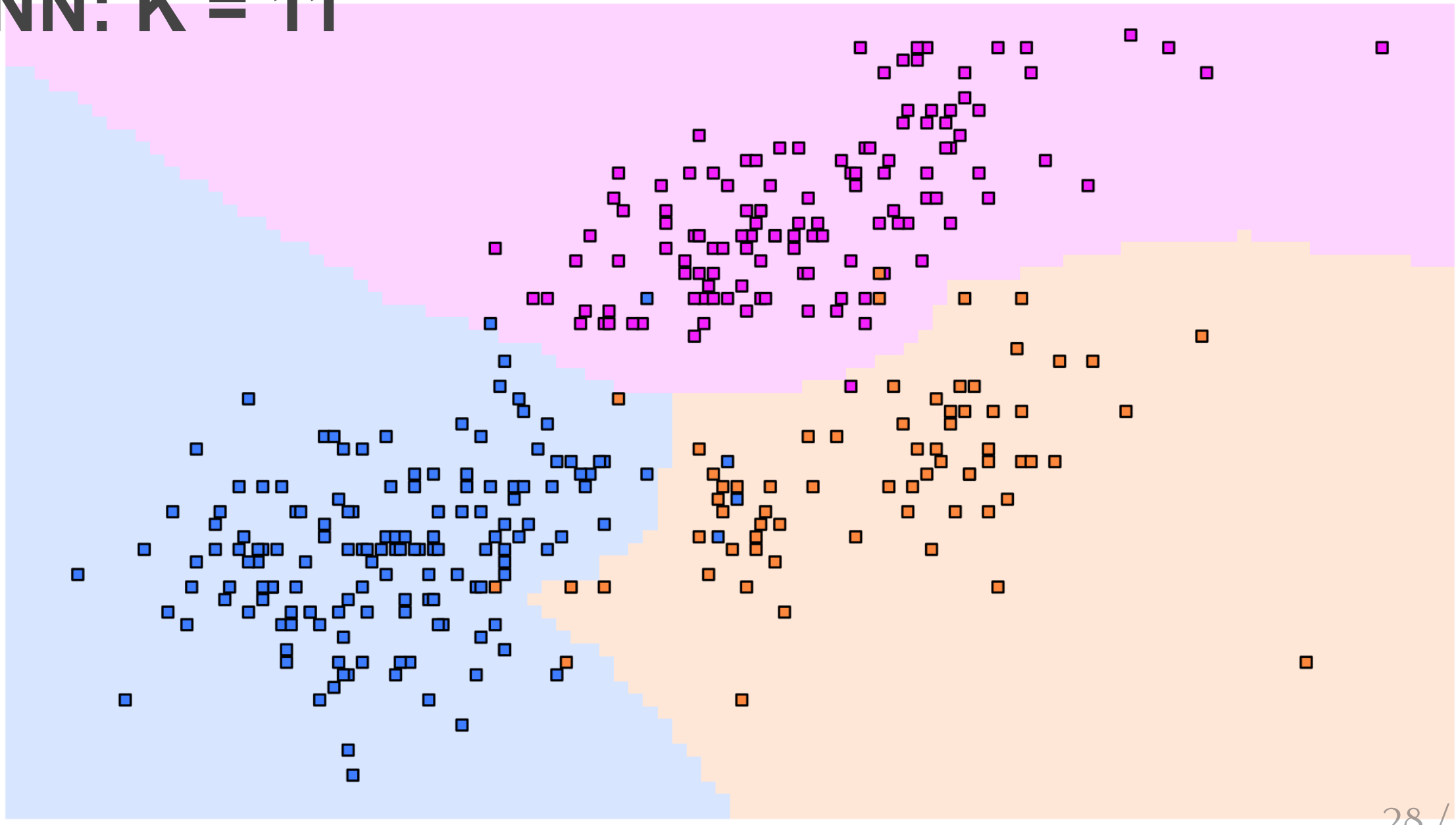
KNN: K = 1



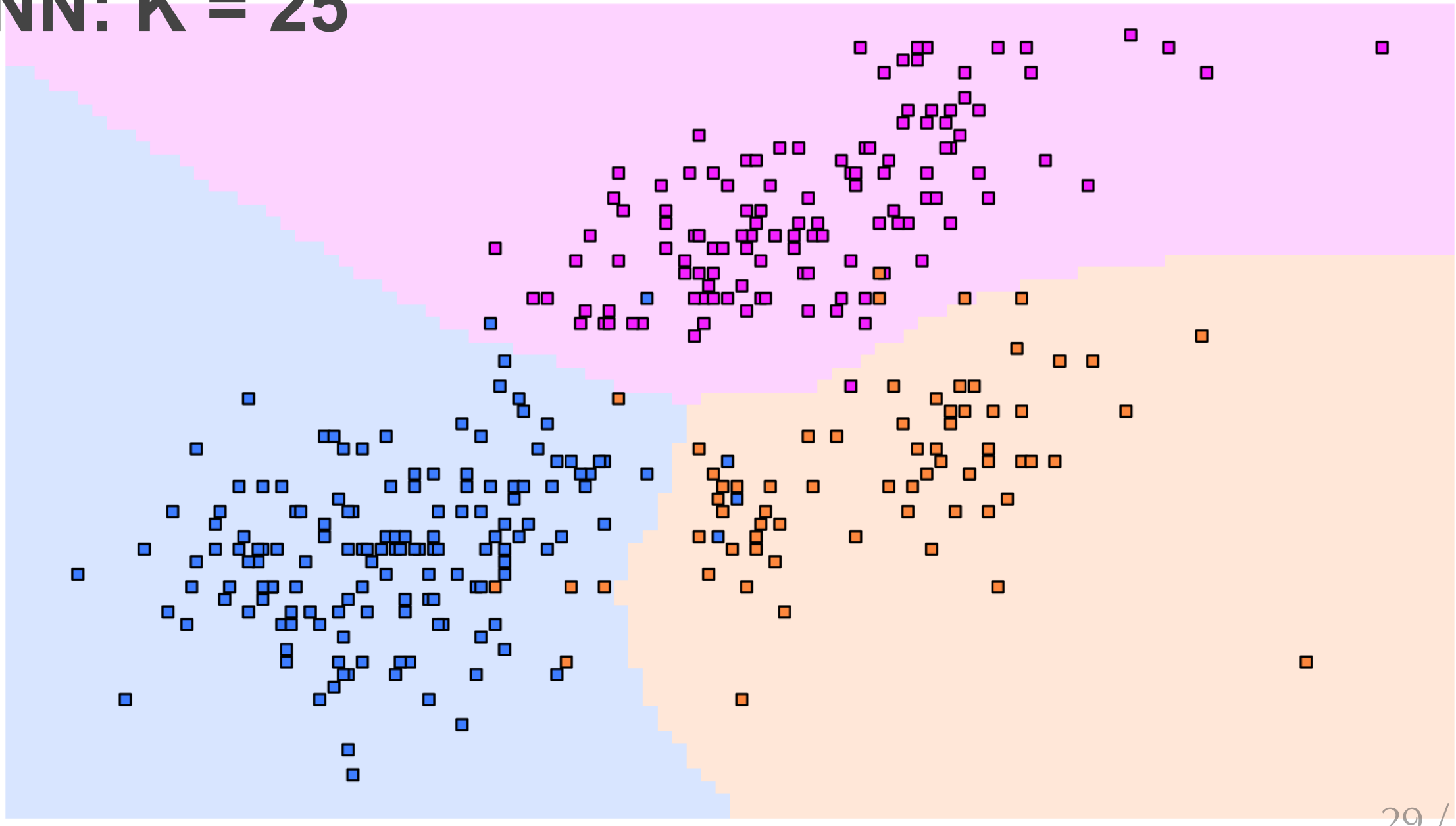
KNN: K = 5



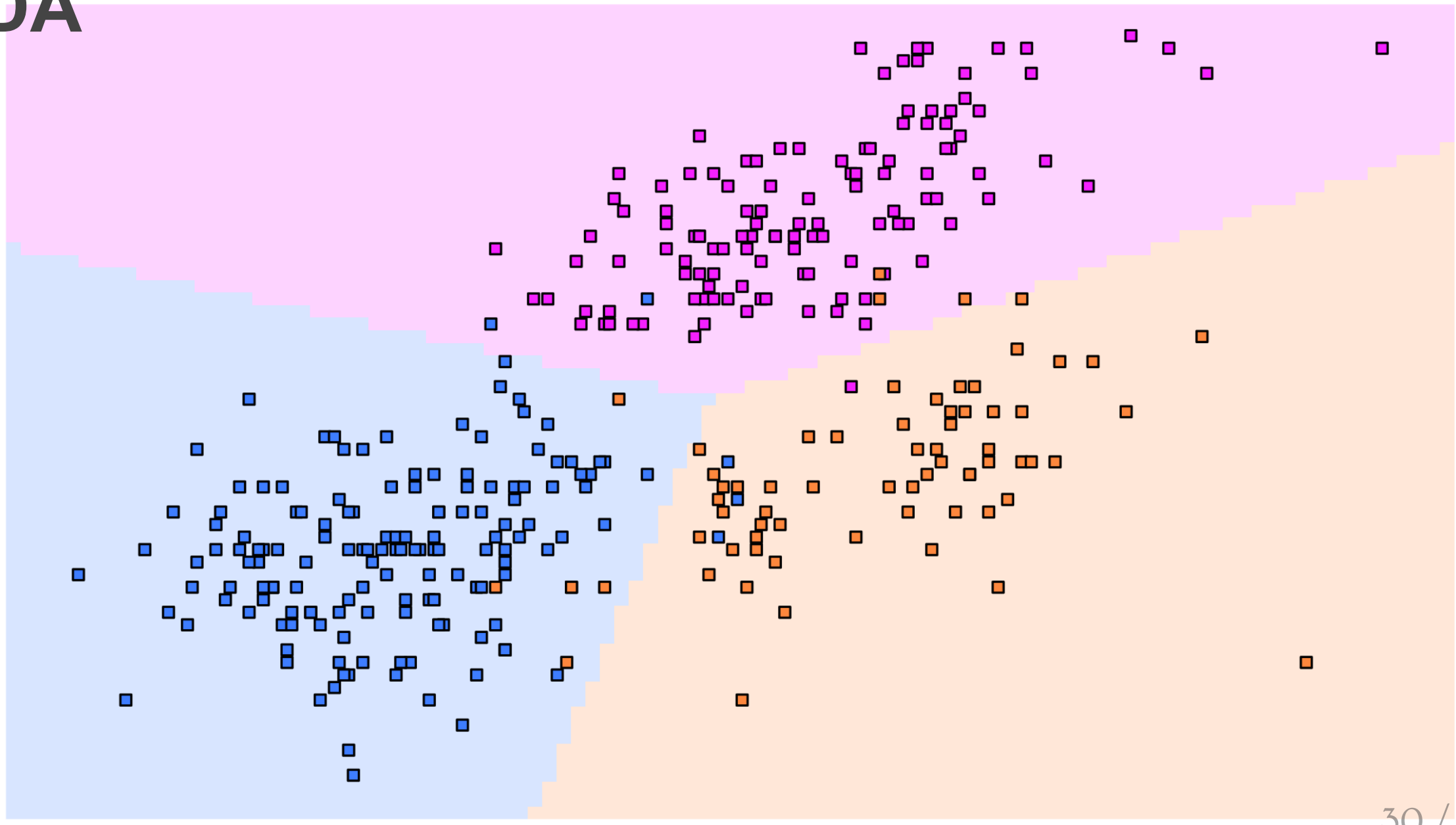
KNN: K = 11



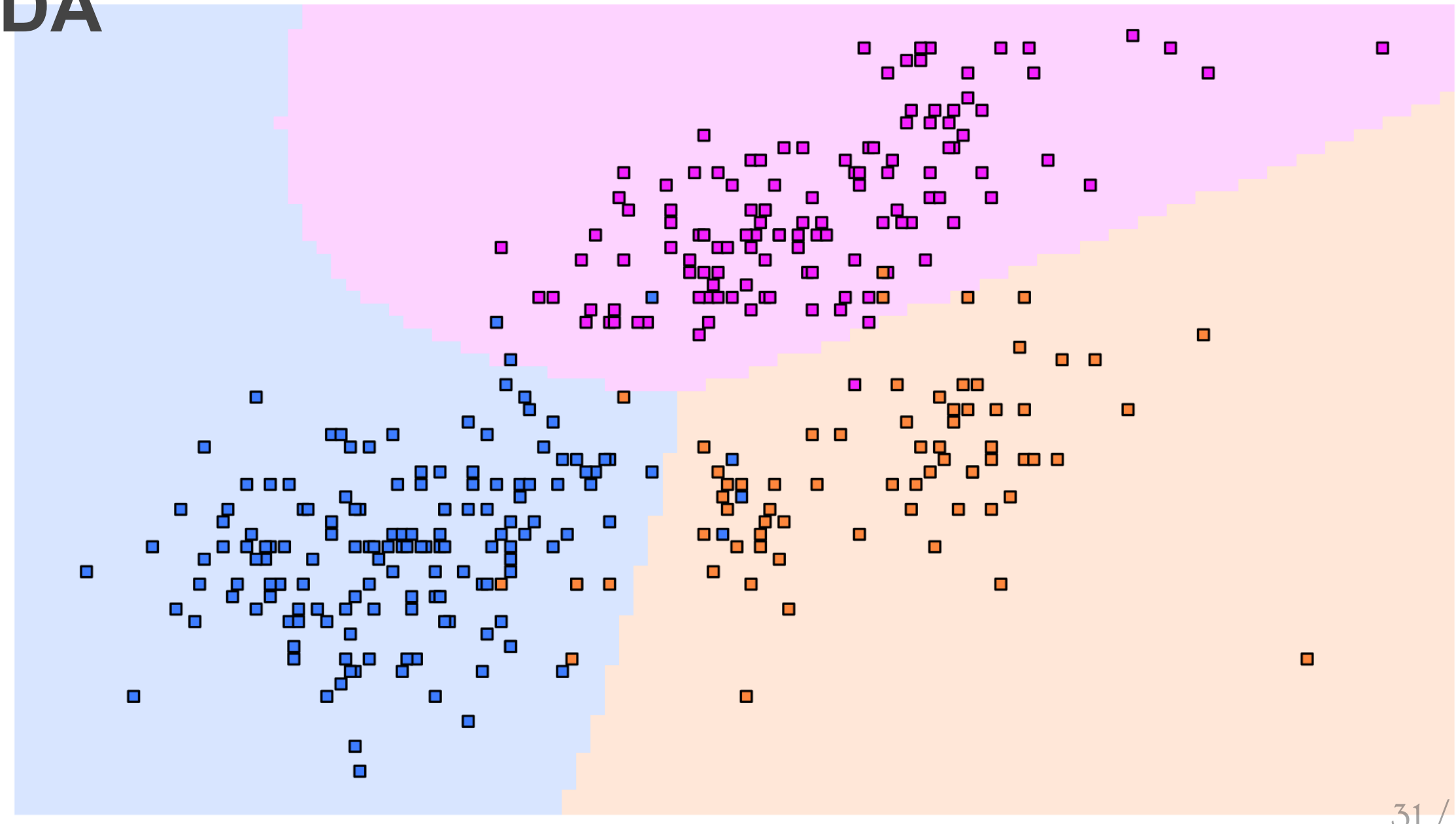
KNN: K = 25



LDA



QDA



KNN

Does not like high-dimensional data

Is VERY flexible

we have to carry around all the data

Here scaling matters!!

KNN regression

KNN can also be used for regression tasks as well by taking a weighted average for the neighbors to give the prediction