KNN regression

For lin. regr we have a really strong global model / assumption about the form of f:

$$y = f(\underset{\sim}{x}) = \underset{\sim}{x}^T \beta$$

e.g. in 1-D

Preds follow a lin. fn. through entire space

Training data affects fit very far away

Benefit: strong global assumption is that it simplifies finding $\hat{f}$ (optimization over p-dim'l space)

(optimization over p-dim'l space)

KNN regr makes a weaker local assumption abat the form of f / $\hat{f}$ so that vals of $\hat{f}$ only depend on nearby training data

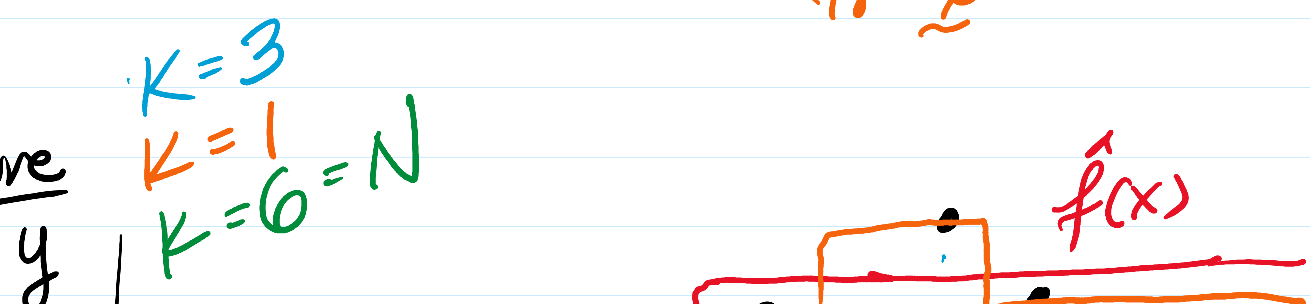
In particular:

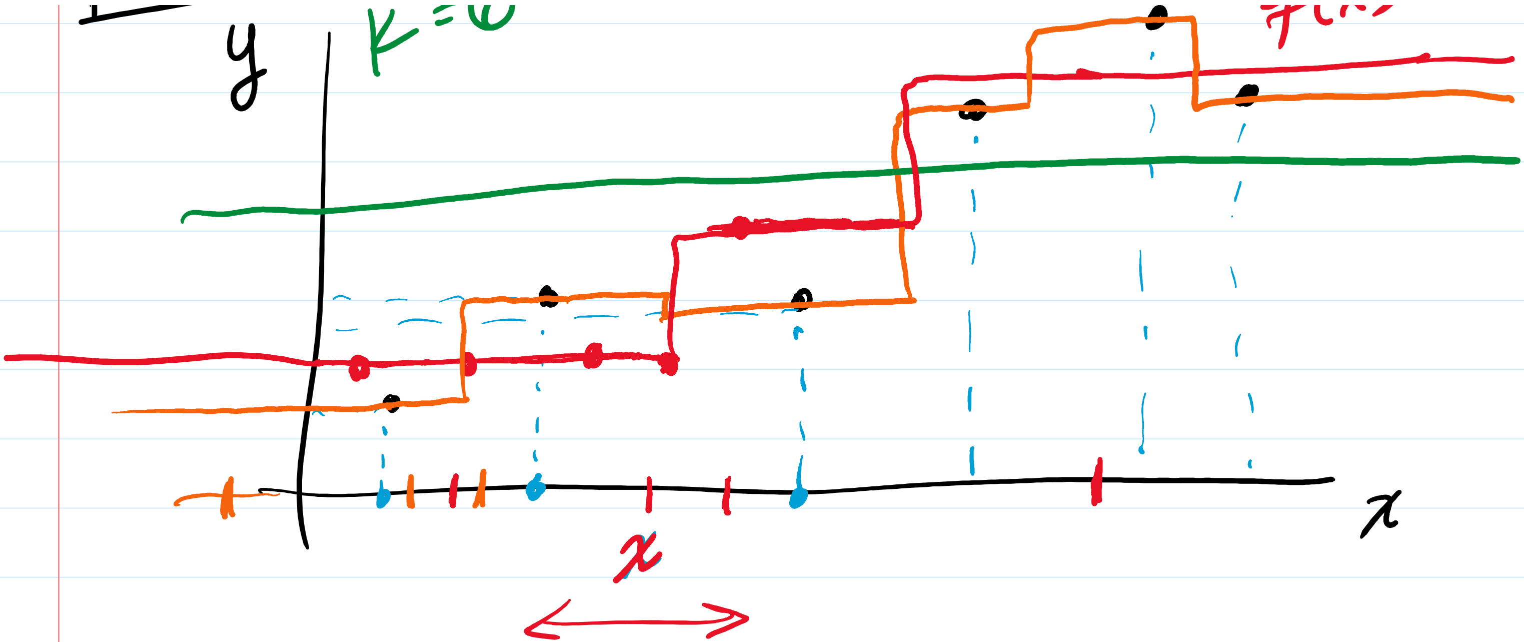
$$\hat{f}(\underset{\sim}{x}) = \frac{1}{K} \sum_{n \in N_K(\underset{\sim}{x})} y_n$$

$N_K(\underset{\sim}{x})$ = set of K nearest traing indices

(those where $\underset{\sim}{x}_n$ closest to $\underset{\sim}{x}$)

picture





General rule: K controls model flexibility

→ how complex $\hat{f}$ is

Small K → very flex model

large K → very inflex model

Model Evaluation : does my $\hat{f}$ perform well?

why? (1) compute perf in its own right

(2) model tuning - engineering feats

(2) <u>model tuning</u> - engineering feats

- tune K for KNN

- choose among competing models

<u>What do we want?</u>

A measure of perf. of $\hat{f}$ on new (unseen/indep) data.

[generalization perf.]

<u>How can we do this?</u>

If we have some data that (at least partially) mimics new data

[evaluation data]

then we:

(1) train on D_{train} to produce $\hat{f} = \hat{f}_{D_{train}}$

(2) ... $\hat{f}$ on D_{eval} ...

(2) eval. $\hat{f}$ on D_{eval}

---

e.g. calculate

- $MSE = \frac{1}{|D_{eval}|} \sum_{(x,y) \in D_{eval}} (y - \hat{f}(x))^2$

- $MAE = \frac{1}{|D_{eval}|} \sum_{(x,y) \in D_{eval}} |y - \hat{f}(x)|$
