Inf2 - Foundations of Data Science: k-Nearest neighbours and evaluation

THE UNIVERSITY of EDINBURGH
informatics

FOUNDATIONS OF DATA SCIENCE

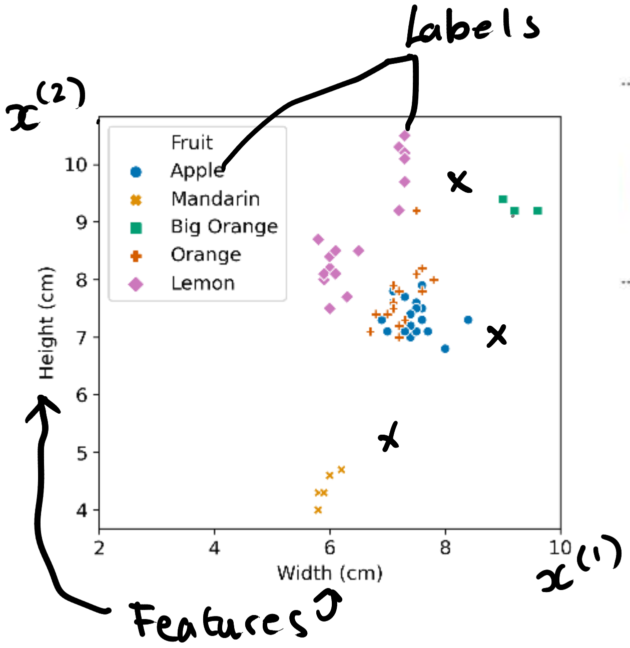
Recap of Monday's lecture

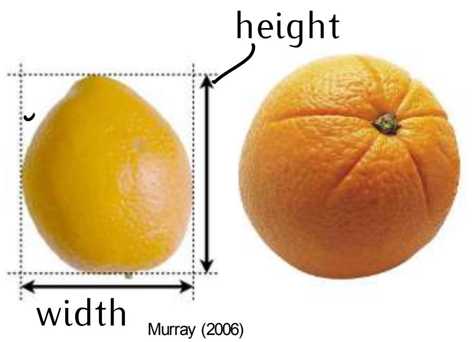
Supervised learning of classifications

Features — Feature vector — Label / class

Income £	Housing	Employment	Repaid
20,000	Rent	Student	Yes
50,000	Owns	Retired	No
⋮	⋮	⋮	⋮
30,000	Rent	Retired	?

Visualising the classification problem





Definition of a classifier

Feature vector $\underline{x} = (x^{(1)}, \ldots, x^{(D)})^T$

$x^{(2)}$, $x^{(1)}$

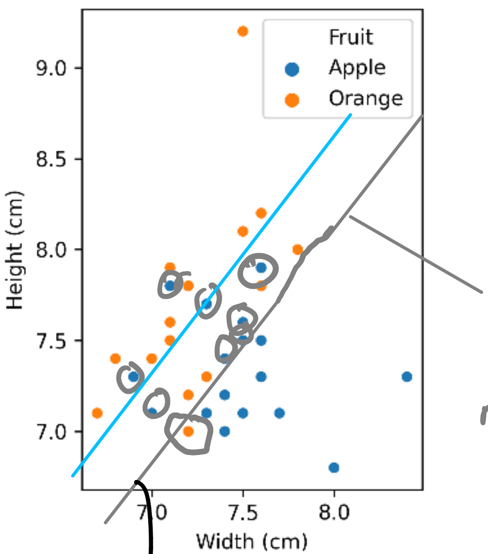
$\underline{x} \rightarrow$ Classifier $\rightarrow c \in \mathcal{C}$

$\mathcal{C} = \{$ Apple, Orange, Lemon, … $\}$

Properties

1. Decision boundaries
2. Classification errors

Constructing a classifier by hand



w_1 w_0

$x^{(2)} > \beta_1 x^{(1)} + \beta_0$

$\Rightarrow$ Orange

Otherwise

$\Rightarrow$ Apple

$x^{(2)} = \beta_1 x^{(1)} + \beta_0$

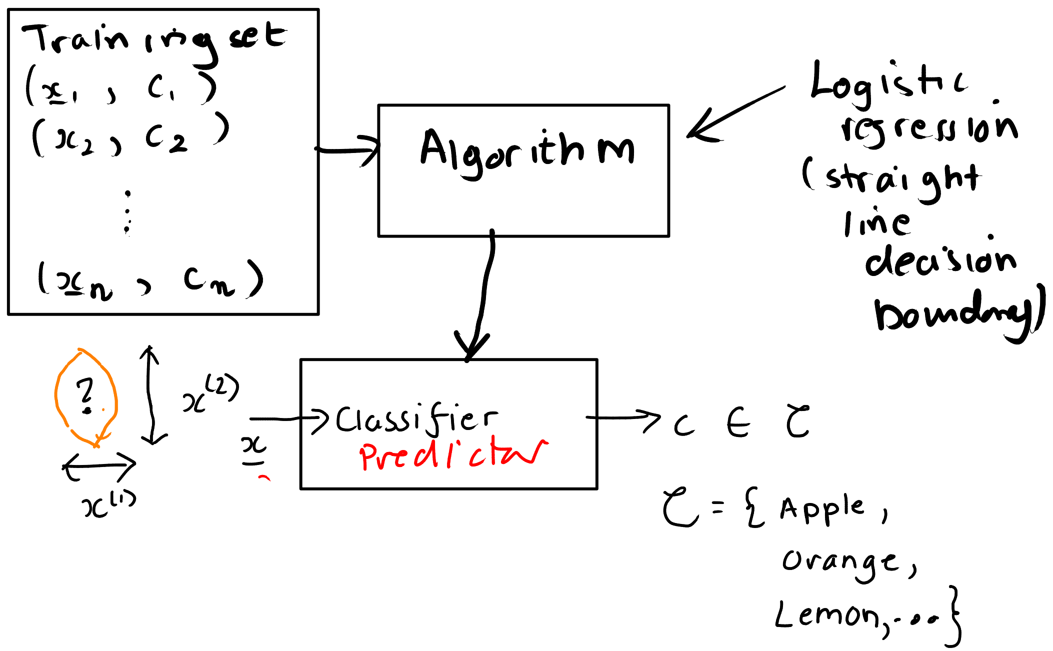
misclassification.

1 + 8 = 9 misclass.

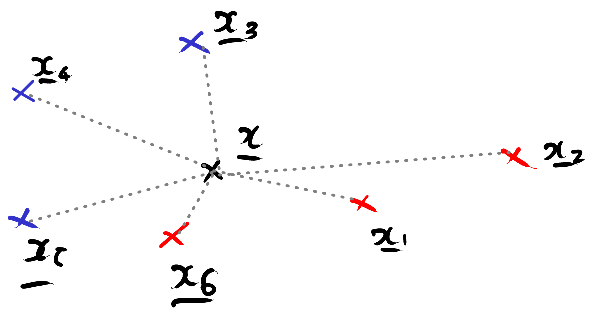
2 + 5 = 7 "

Decision boundary

Constructing a classifer by supervised learning



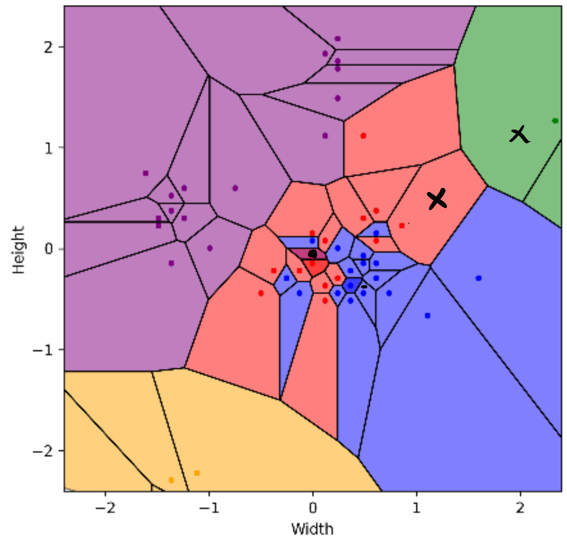
Principle of nearest neighbour classification



Euclidean distance

$$d(\underline{x}, \underline{x}_i) = \sqrt{(x_i^{(1)} - x^{(1)})^2 + (x_i^{(2)} - x^{(2)})^2}$$

Nearest neighbour classification of fruit



- Standardised variables
- Often better for distance-based algorithms

Evaluating generalisation to unseen data

Training set 70%

Test set 30%

Data

1-NN

Training set error ≈ 0

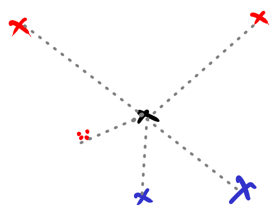
Test set error > 0

Never use the test set to train the classifier and then quote the error based on the test set
The estimate of the error will be too optimistic.

Principle of k-NN

To improve generalisation look at...

k- nearest neighbours

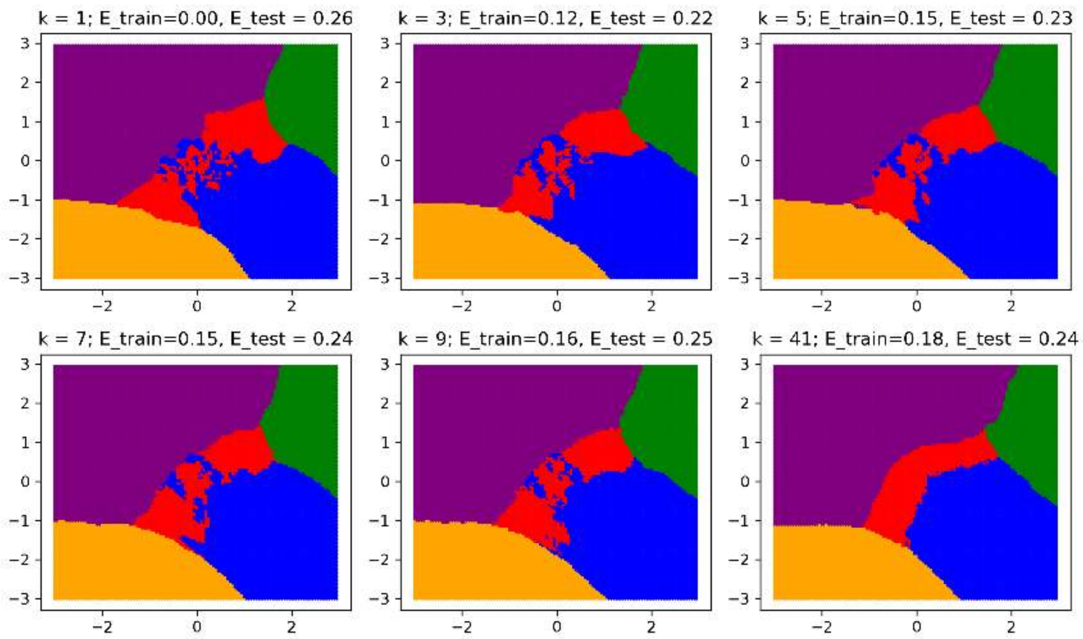


k	R	B	Winner
1	:	0	R
2	1	1	B/R
3	1	2	B
4	2	2	B/R
5	3	2	R

Ties: random resolution or weighting

Odd numbers of k

Effect of k-NN

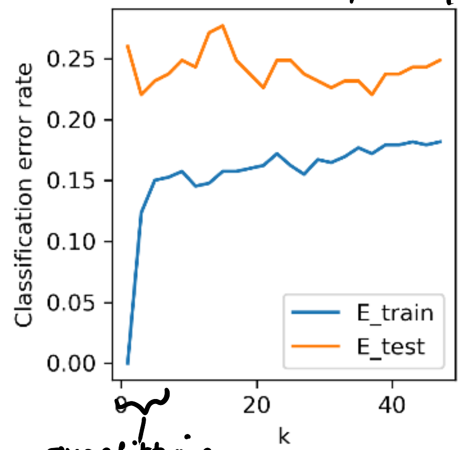


How to pick k?

k is a hyperparameter

under fitting

over-generalisin



overfitting

under-generalising

Overview

- Metrics

- k-NN regression

- Evaluation - K-fold cross validation

Inf2 - Foundations of Data Science: k-Nearest neighbours and evaluation Metrics

THE UNIVERSITY of EDINBURGH
informatics

FOUNDATIONS OF DATA SCIENCE

(Classification) accuracy

Definition: Classification accuracy

$= 1 -$ error rate

$$= 1 - \frac{\#\text{ items misclassified}}{\#\text{ items}}$$

Accuracy and unbalanced classes

Definition =

Question: You want to test for a disease that you think around 5% of the population have. A company offers you a test that has 95% accuracy. Should you use it?

How to build a 95% accurate Covid classifier

Better measures for unbalanced classes

1. Sensitivity = Fraction of positives classified as positive

2. Selectivity = Fraction of negatives classified as negative

If only 5% of cases are truly positive then classifying all cases as negative gives:

Sensitivity : 0%
Selectivity : 100%
Accuracy : 95%

Confusion matrix

Predicted class

		P	N
Actual class	Positive P		
	Negative N		

Bad classifier

	P	N

Accuracy =

Sensitivity =

Selectivity =

Other metrics for classifiers

- Precision and recall
- F1
- In the context of a classifier with a adjustable threashold (e.g. Logistic regression, see later lectures)
  - Receiver operator characteristic
  - Area under the curve

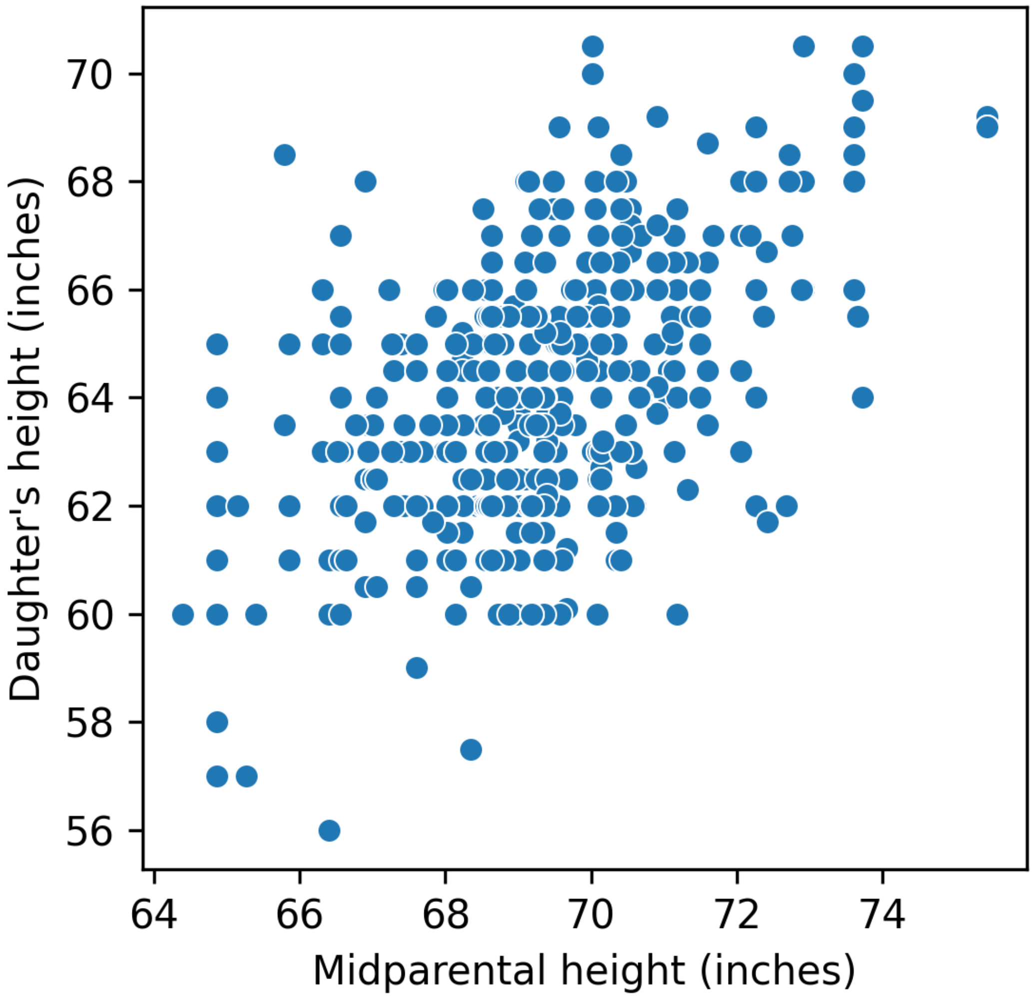
Inf2 - Foundations of Data Science: k-Nearest neighbours and evaluation - k-NN regression

THE UNIVERSITY of EDINBURGH
informatics

FOUNDATIONS OF DATA SCIENCE

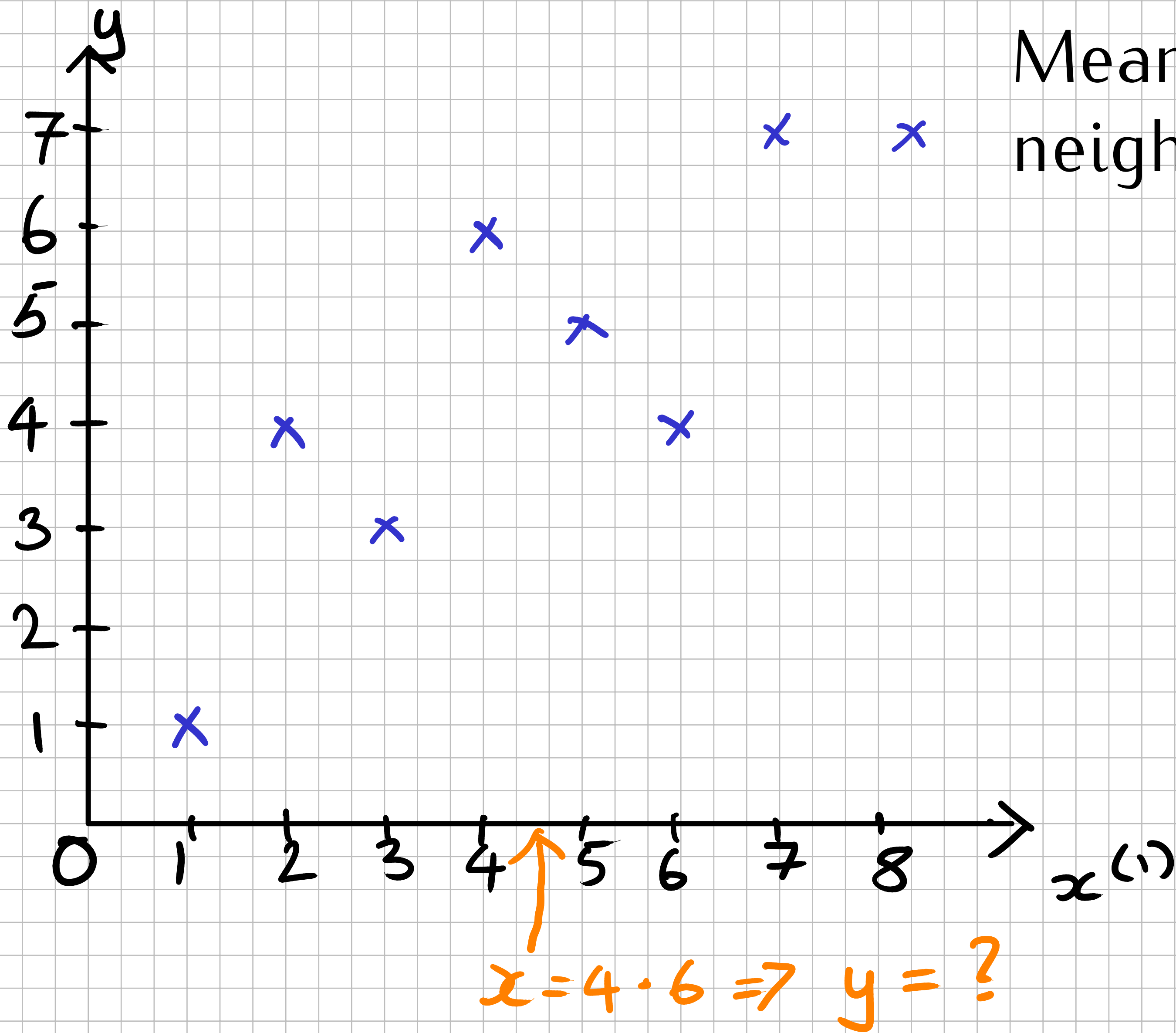
k-NNs for regression

Height of daughter versus parent height



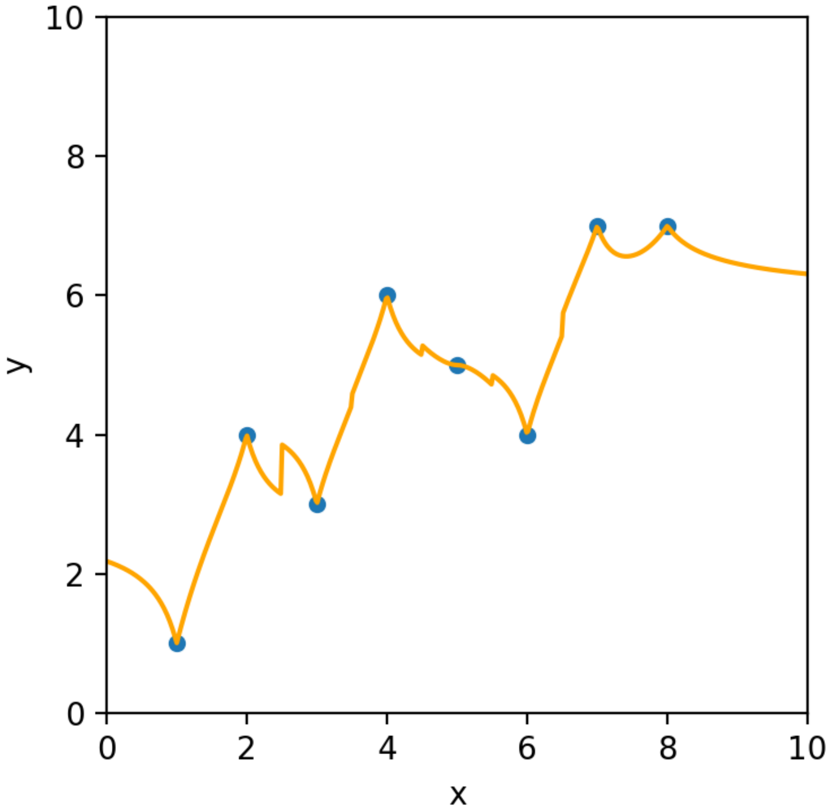
Could we use the principle of k-Nearest Neighbours to predict the daughter's height from the midparental height?

1-D example

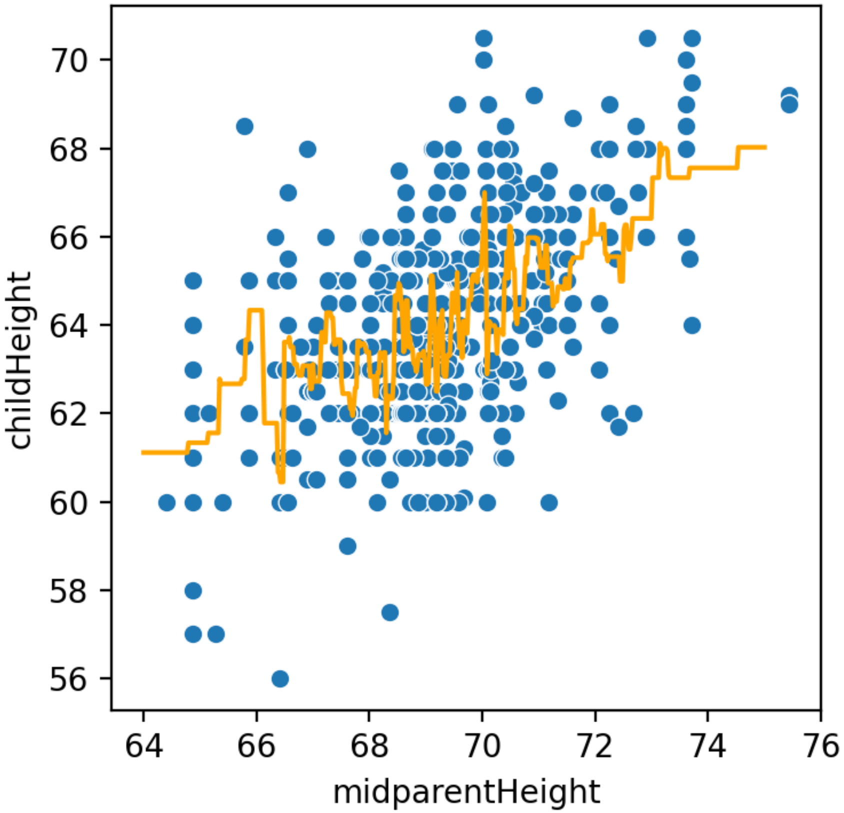


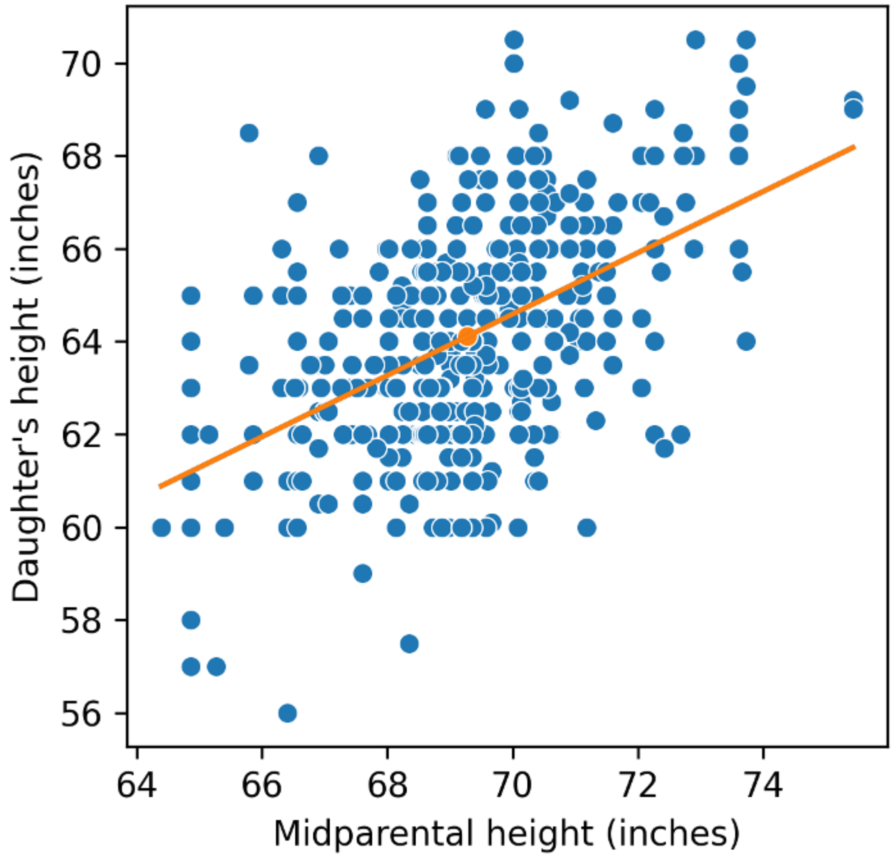
Mean of 3-nearest neighbours

1-D example weighted by distance

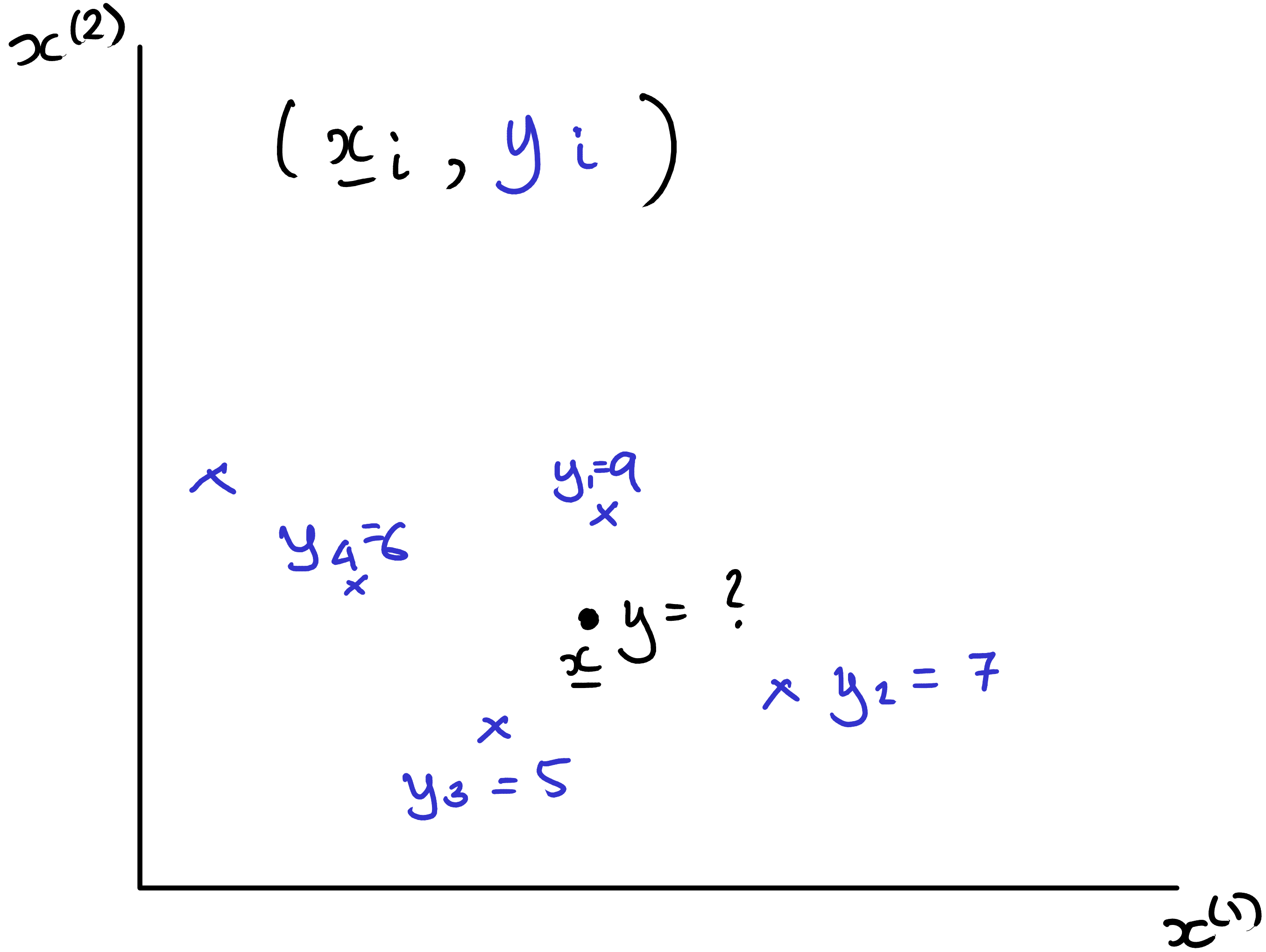


k-NN regression versus linear regression





k-NN in more than one dimension



N_k - set of k points nearest to $\underline{x}$

Uniform weights:

$$y = \frac{1}{k} \sum_{i \in N_k} \underline{x}_i$$

Inverse-distance weighted

$$y = \frac{\sum_{i \in N_k} 1/d(\underline{x}, \underline{x}_i)\, y_i}{\sum_{i \in N} 1/d(\underline{x}, \underline{x}_i)}$$

Why use k-NN regression?

Metrics for regression

Same as for Linear regression!

- (Root) Mean squared error

- R-squared

Also:

- Mean Absolute Error (MAE)

Inf2 - Foundations of Data Science: k-Nearest neighbours and evaluation - K-fold cross-validation

THE UNIVERSITY of EDINBURGH
informatics

FOUNDATIONS OF DATA SCIENCE

The supervisedmachine learning paradigm

.

1. Split data into training, validation and test sets
2. Use training and validation data to select
   (a) an alogorithm
   (b) hyperparameters (if applicable)
3. Use test data to report performance of resulting predictor (classifier or regressor)

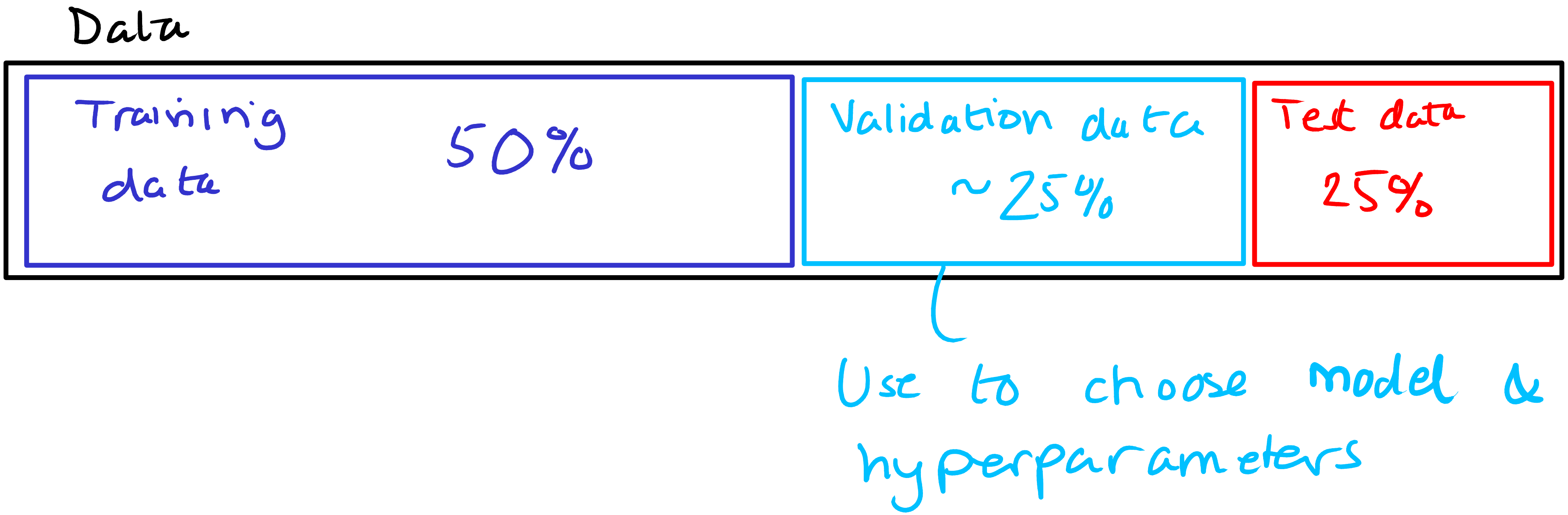
Limitations of the supervised machine learning paradigm

Data can change over time
- e.g. Covid symptoms changed over time

Selection of training, validation and test sets
- should be drawn from the same distribution
- should represent the real world to which the predictor will be applied - c.f. Ethics!

Limited amount of data

Training, validation and test data

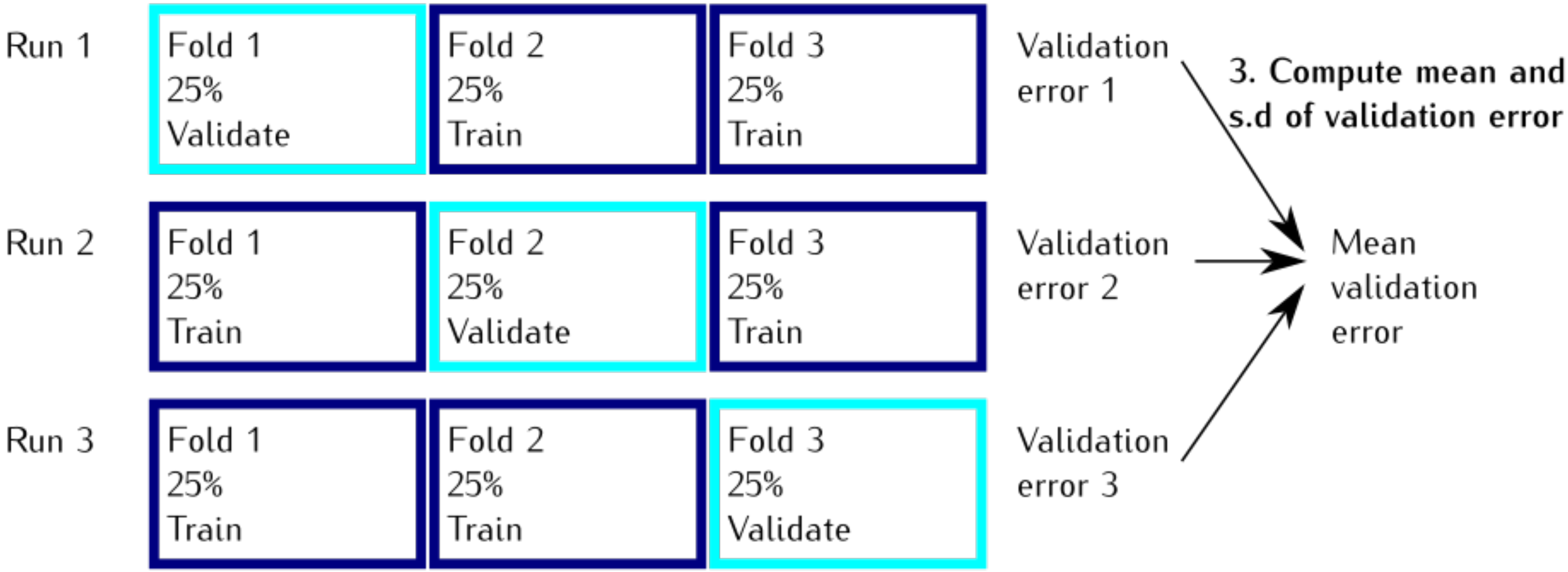


Problem: with small data, we're not able to get a very good estimate fo the validation loss/error

Solution: K-fold cross-validation

1. Split into testing data and non-testing data. Spilt non-testing data into 3 folds.

2. For each model and hyperparameter, train and compute validation error three times:



4. Pick hyperparameter with lowest mean validation error. Train on all non-testing data and report performance on test data.

Model-fitting procedure

1a. Split data into training and testing sets

1b. Split training data into K folds

2. for model in models

    for hyperparameter in hyperparameters

        for k in K

            train model with hyperparameter on all folds apart from k

            test trained model on k^{th} fold

        store mean and s.d of test errors with hyperpar.

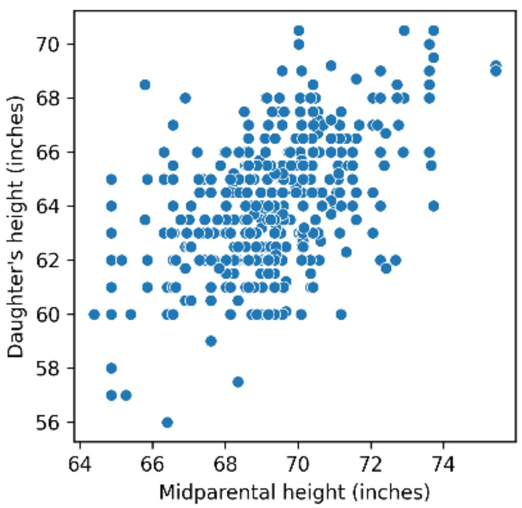
3. Choose model and hyperparameters with best mean test error

Reporting test metric

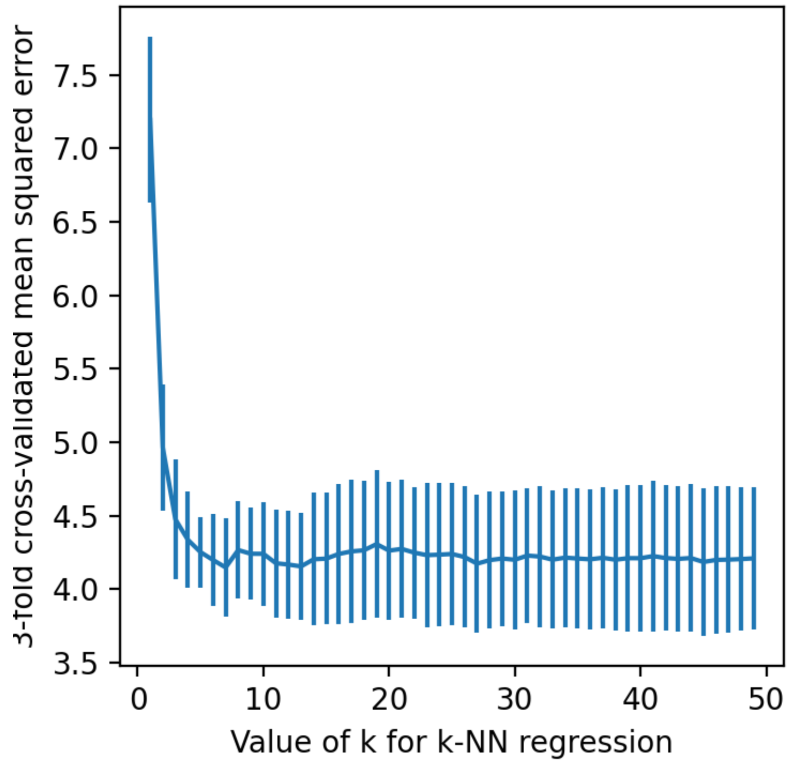
- Although validation metric gives estimate of expected test metric, still need to compute metric on separate test set

- Exception: in a competition, the organisers create training and testing sets, keeping test set secret

Example - applying cross-validation with k-NN vs linear regression



k-NN cross-validation

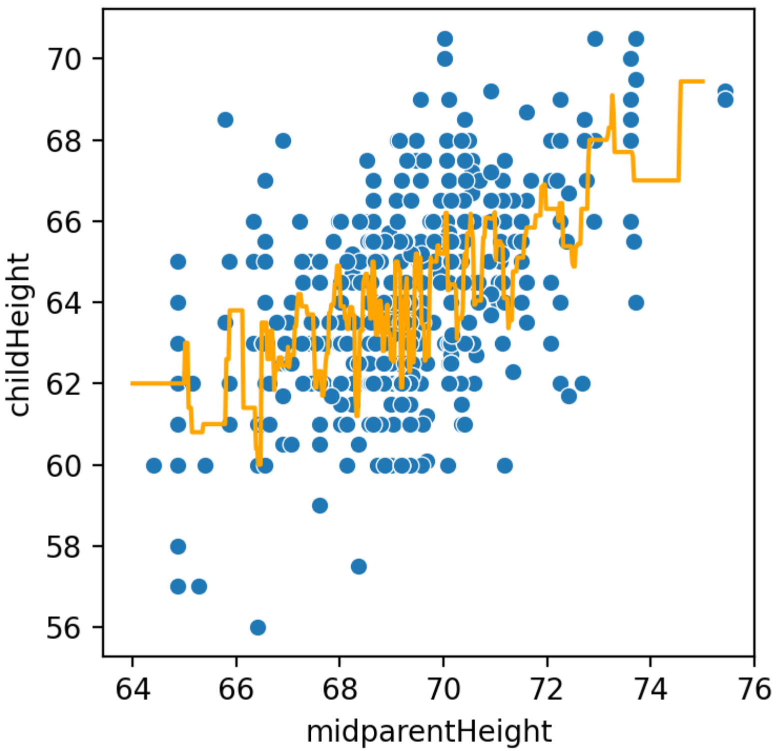


Linear regression: MSE mean (s.d) = 3.95 (0.21)

Test results

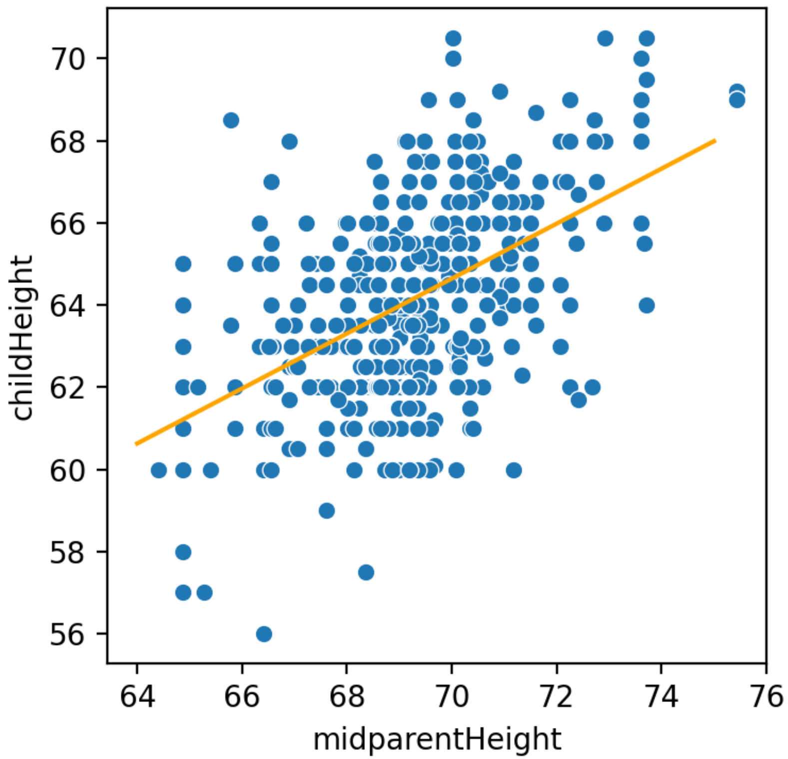
k-NN

MSE = 4.87 sq in
RMSE = 2.21 in
MAE = 1.68 in



Linear regression

MSE = 4.56 sq in
RMSE = 2.14 in
MAE = 1.66 in



Summary

- Classification metrics: careful choosing what to measure
  - accuracy can be misleading with unbalanced classes

- k-NN regression good for predicting nonlinear relationships between a numeric label and numeric predictors/features
  - Often worth trying linear regression!

- Cross-validation helps choose ML algorithms and hyperparameters

- Test data is to estimate how the predictor works in the real world - not for training!!