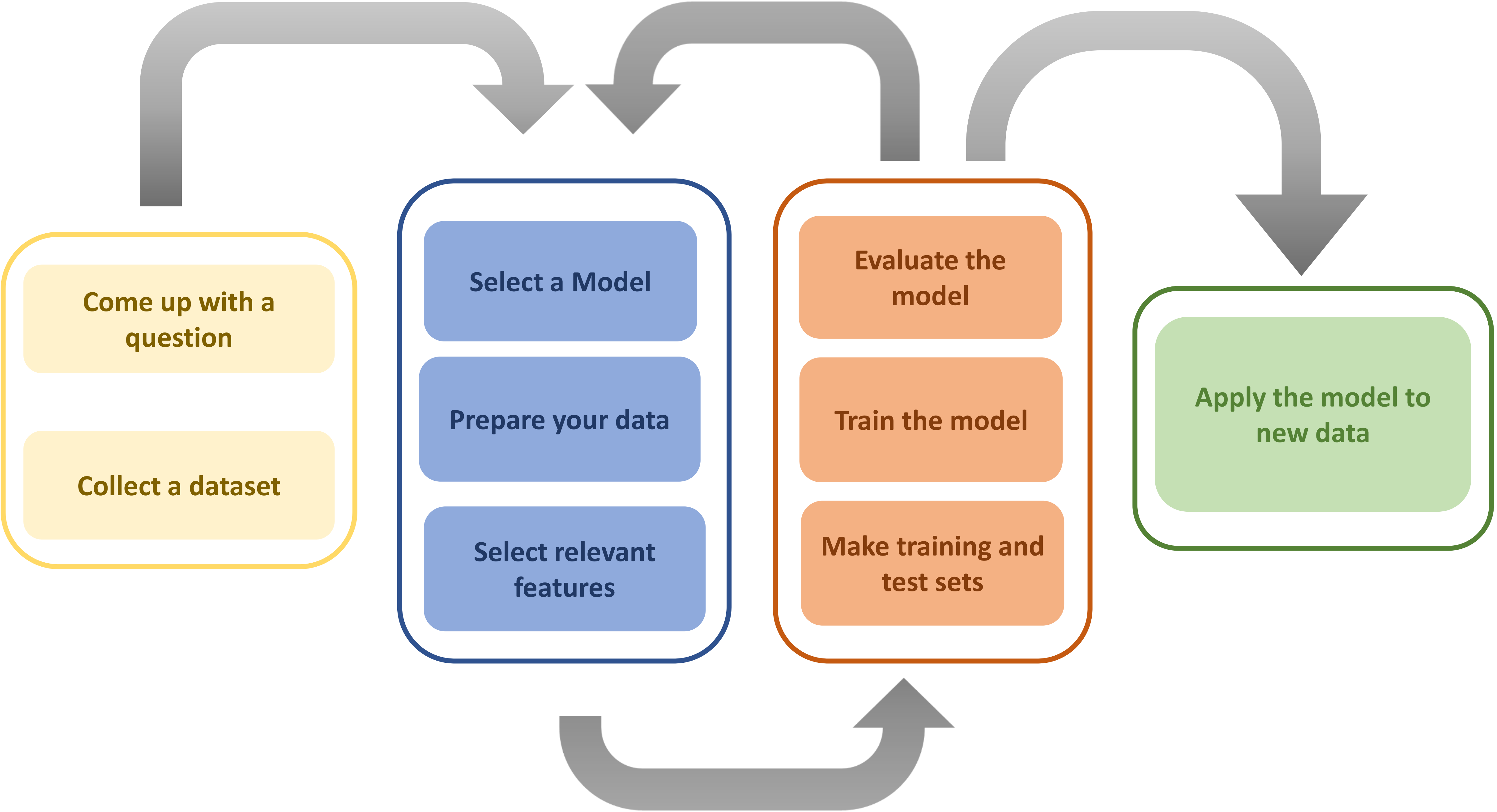
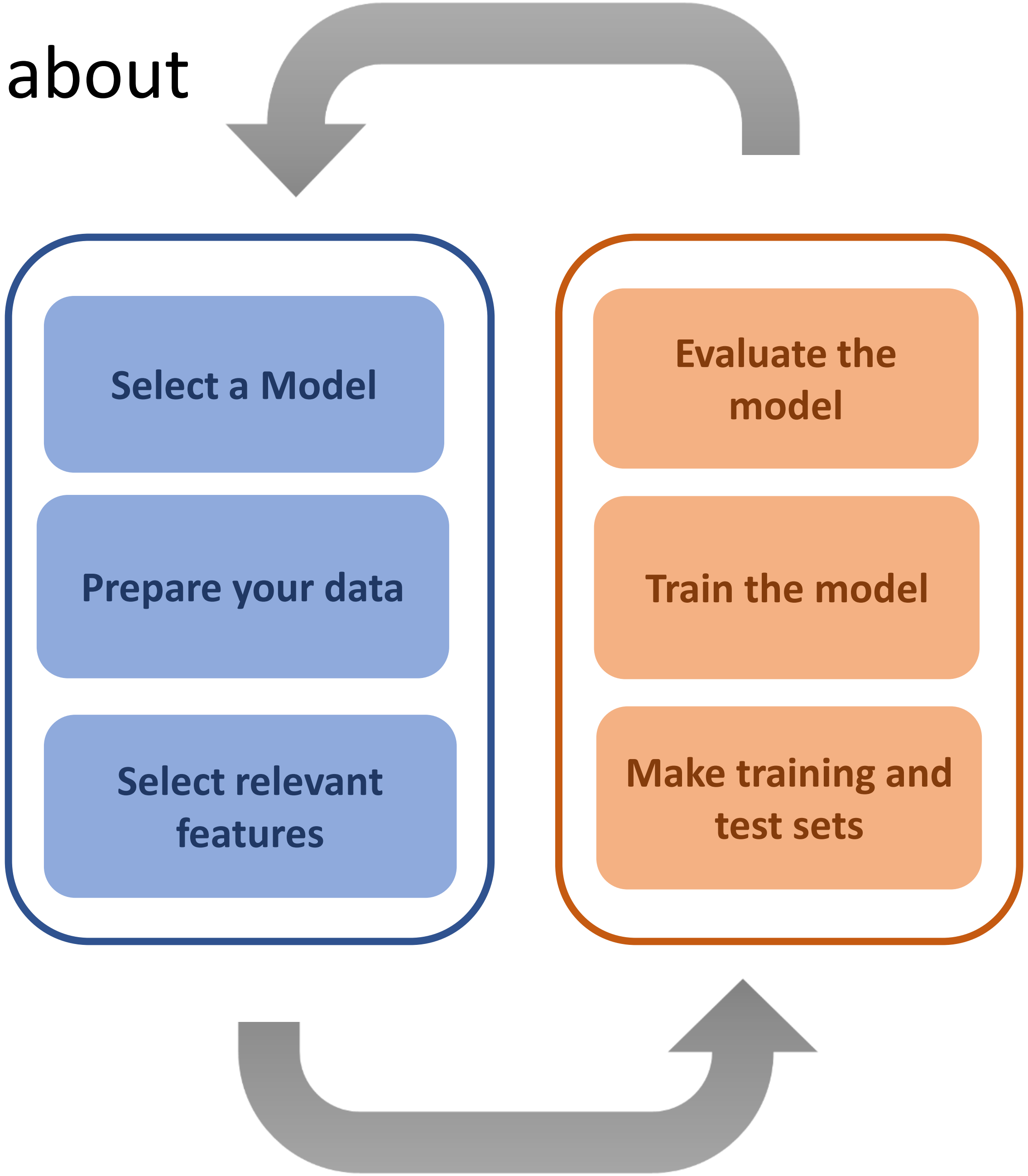
Machine Learning Fundamentals

Disclaimer

This summary is by no means extensive. It is restricted to the contents of this course so far and meant as a very cursory overview.


Come up with a question
Collect a dataset
Select a Model
Prepare your data
Select relevant features
Evaluate the model
Train the model
Make training and test sets
Apply the model to new data

What this course is about



Example

I am planning to sell my apartment but don‘t know how much money to expect for it.

Come up with a question

Collect a dataset

Collect a dataset of houses sold in the past

Area	Bedrooms	Distance to city center	Construction	Price
78 m^2	3	10,1 km	1963	112.000 €
121 m^2	4	6,3 km	2001	243.200 €
43 m^2	1	3,0 km	1983	51.000 €
68 m^2	2	17,2 km	1952	43.500 €
92 m^2	3	12,4 km	2010	82.400 €

Example

Features

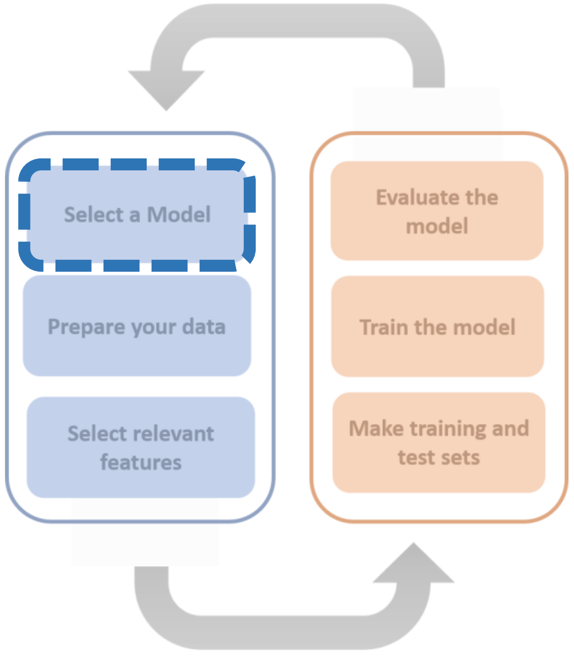
The number of features is the dimensionality of your data (here 4)

Target

Datapoint:

This entire set has 5 datapoints

Area	Bedrooms	Distance to city center	Constuction	Price
78 m^2	3	10,1 km	1963	112.000 €
121 m^2	4	6,3 km	2001	243.200 €
43 m^2	1	3,0 km	1983	51.000 €
68 m^2	2	17,2 km	1952	43.500 €
92 m^2	3	12,4 km	2010	82.400 €

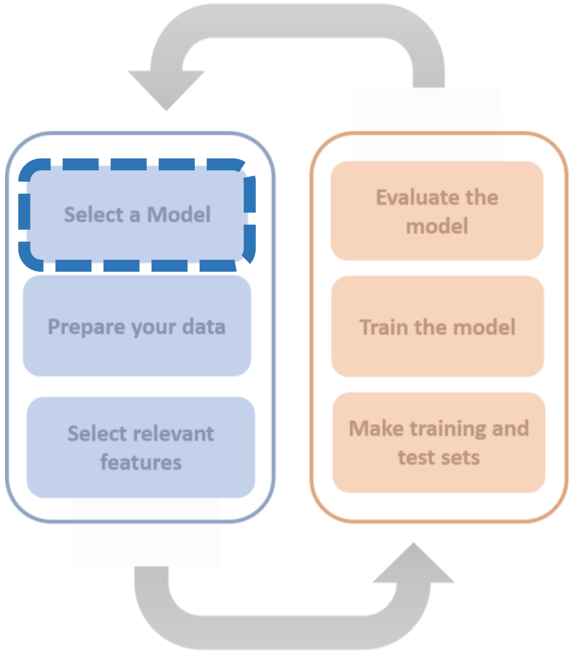


Model selection

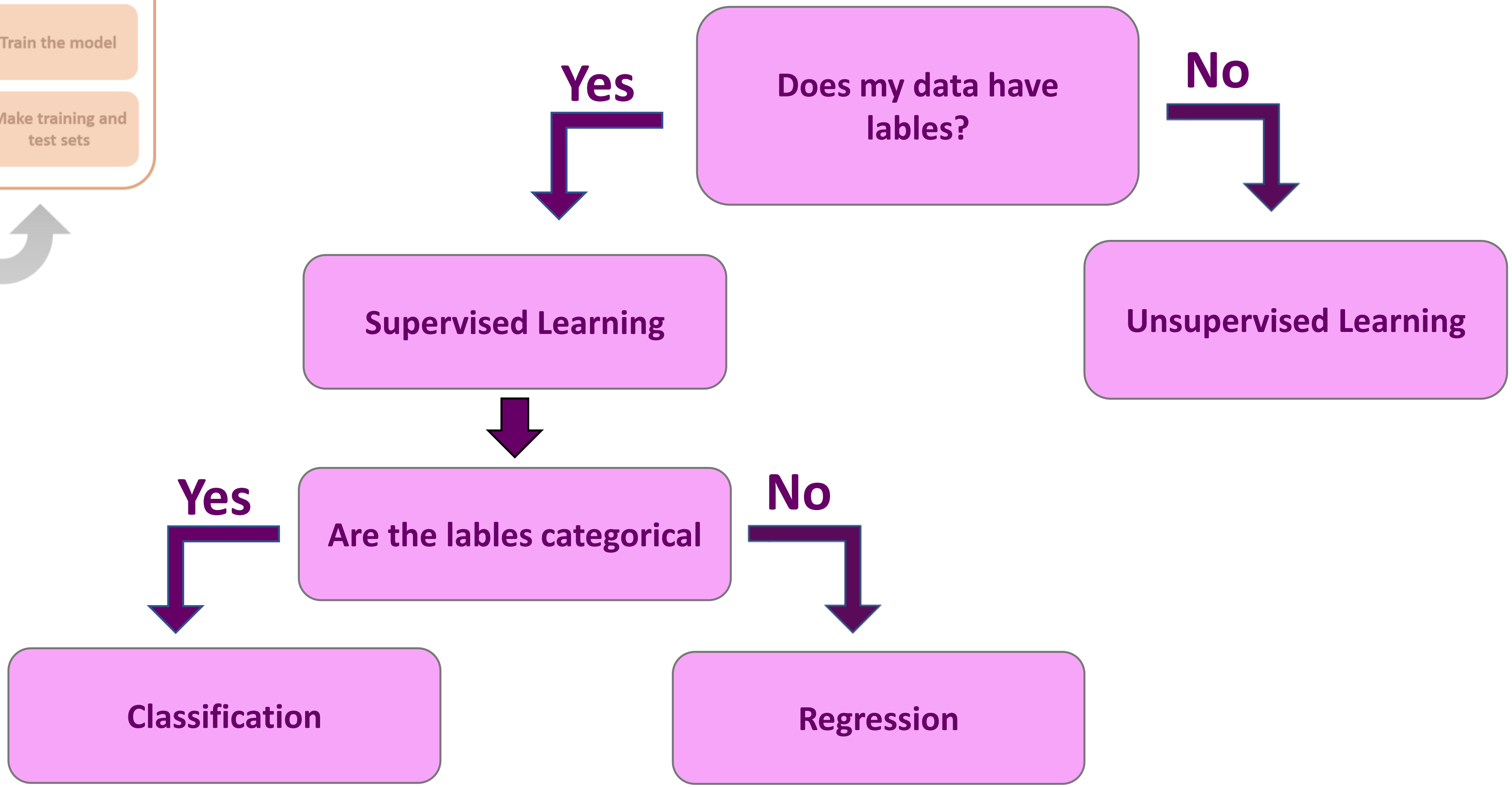
Identify the Type of Problem

Select a Model Type

Choose the Model Parameters



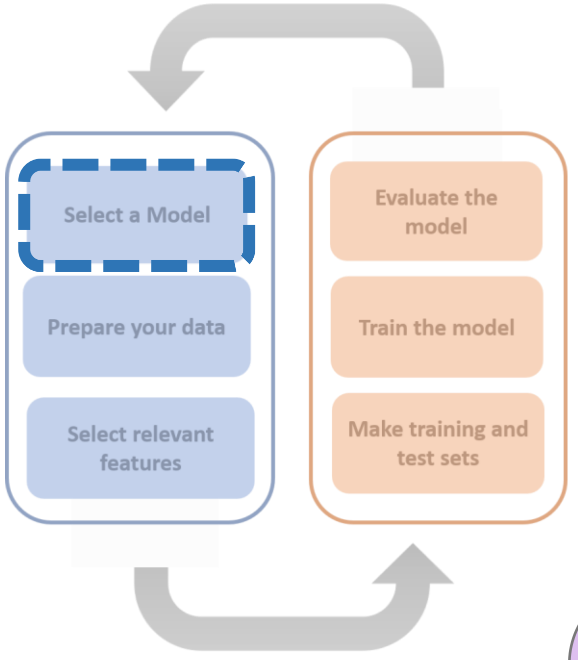
Identify the Type of Problem



Example

Continuous target, so it‘s a regression problem!

Area	Bedrooms	Distance to city center	Constuction	Price
78 m²	3	10,1 km	1963	112.000 €
121 m²	4	6,3 km	2001	243.200 €
43 m²	1	3,0 km	1983	51.000 €
68 m²	2	17,2 km	1952	43.500 €
92 m²	3	12,4 km	2010	82.400 €



Select the Model Type

Classification

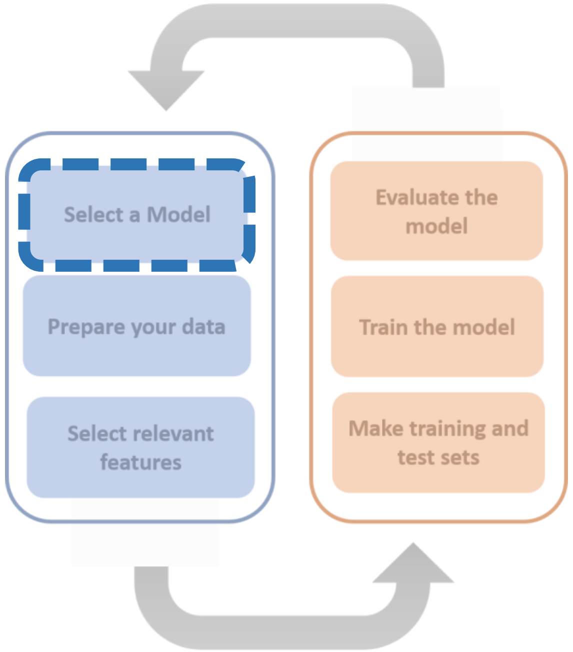
- Logistic Regression
- Decision Tree Classifier
- Random Forrest Classifier
- Ada Boost Classifier
- Gradient Boost Classifier

Regression

- Linear Regression
- Polynomial Regression
- Decision Tree Regressor
- Random Forrest Regressor
- Ada Boost Regressor
- Gradient Boost Regressor

Unsupervised Learning

- k- means clustering
- Hierarchical Claustering
- DBSCAN



Choose the model parameters

Trees

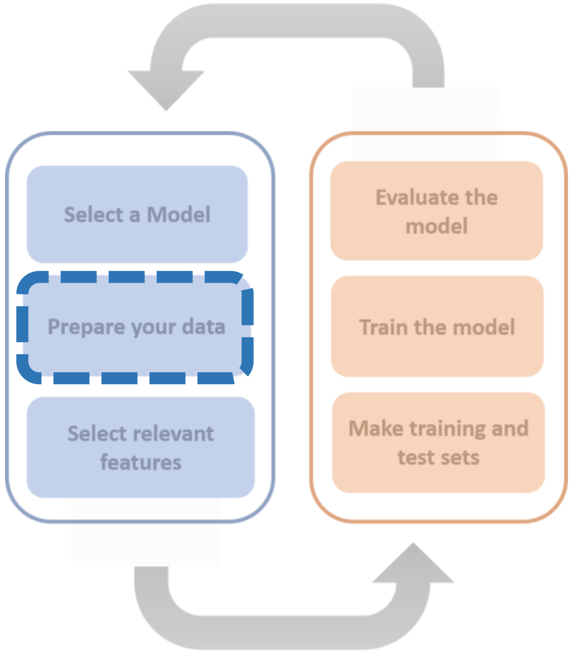
- Tree depth
- Gini or Entropy

Ensemble

- Number of estimators
- Type of estimators

Clustering

- Number of Clusters



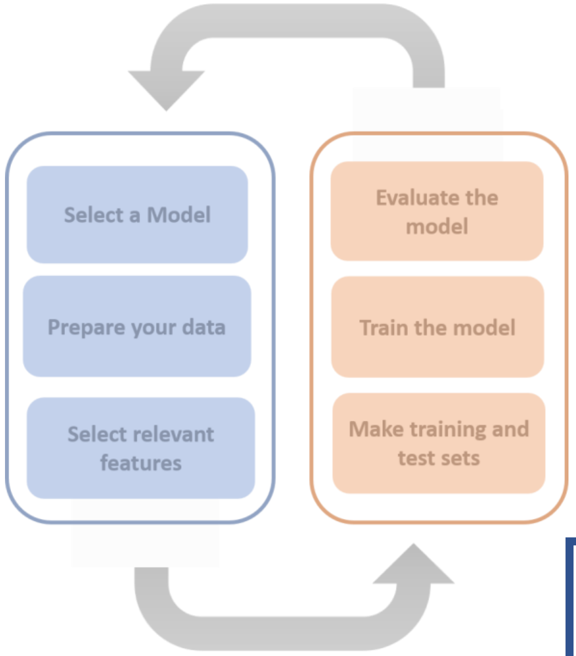
Prepare the data

- Check for missing data and outliers
- Make sure the data is on a reasonable scale (Scaling and Normalization)
- Encode categorical data

Color
Red
Blue
Yellow
Blue
Red

Color
1
2
3
2
1

Color_r	Color_b	Color_y
1	0	0
0	1	0
0	0	1
0	1	0
1	0	0

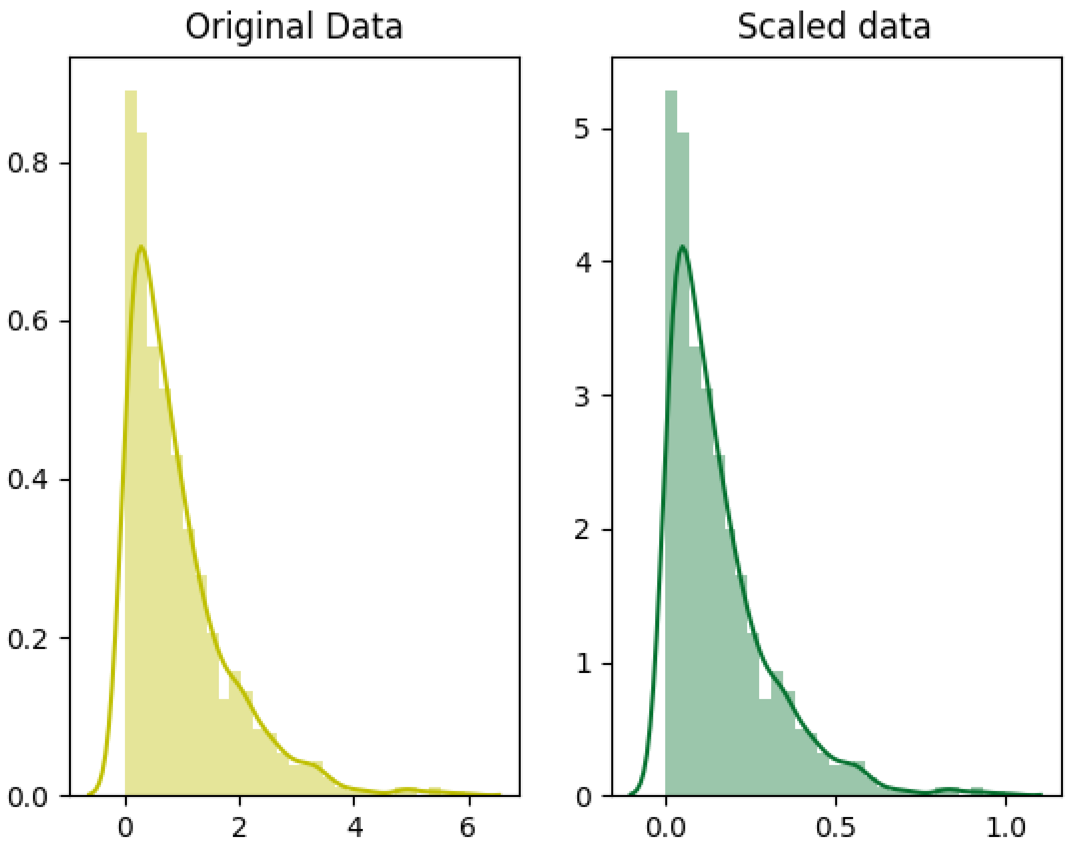


Scaling

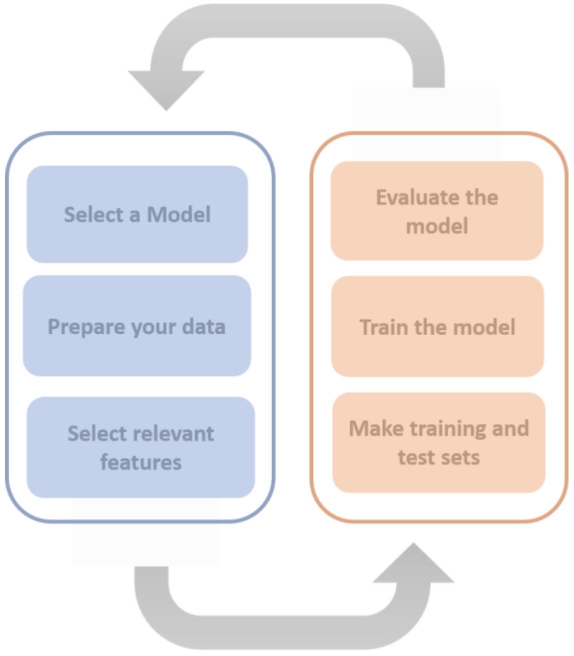
Change the range but not the distribution.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Important for algorithms that calculate distnaces (like k-means or SVM)



https://kharshit.github.io/blog/2018/03/23/scaling-vs-normalization



Standardization/Normalization

$$x' = \frac{x - x_{mean}}{\sigma}$$

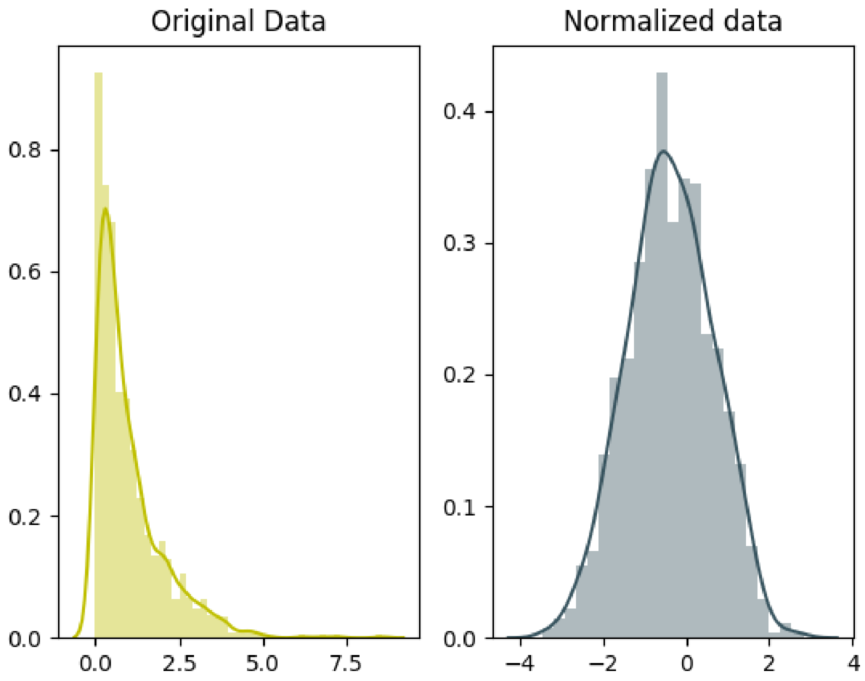
Normalization

Creates a distribution with mean 0 and variance 1

$$x' = \frac{x - x_{mean}}{x_{max} - x_{min}}$$

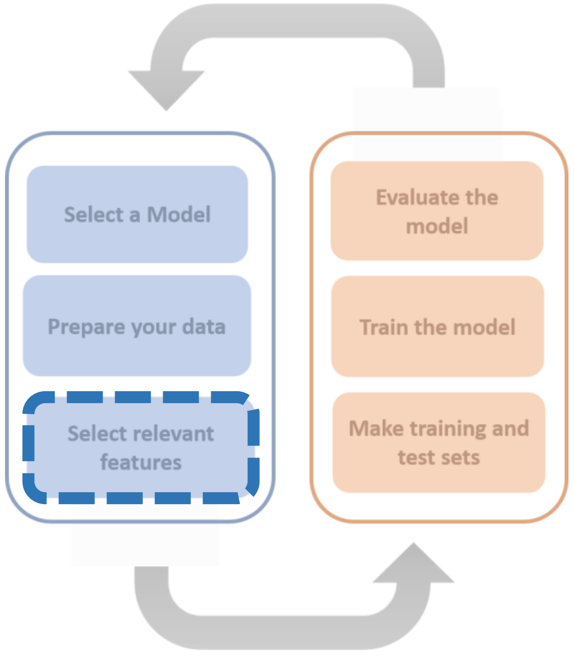
Standardization

Creates a distribution values in [0,1]



You need to normalize our data if you're going use a machine learning or statistics technique that assumes that data is normally distributed e.g. t-tests, ANOVAs, linear regression, linear discriminant analysis (LDA) and Gaussian Naive Bayes.

https://kharshit.github.io/blog/2018/03/23/scaling-vs-normalization



Select relevant features

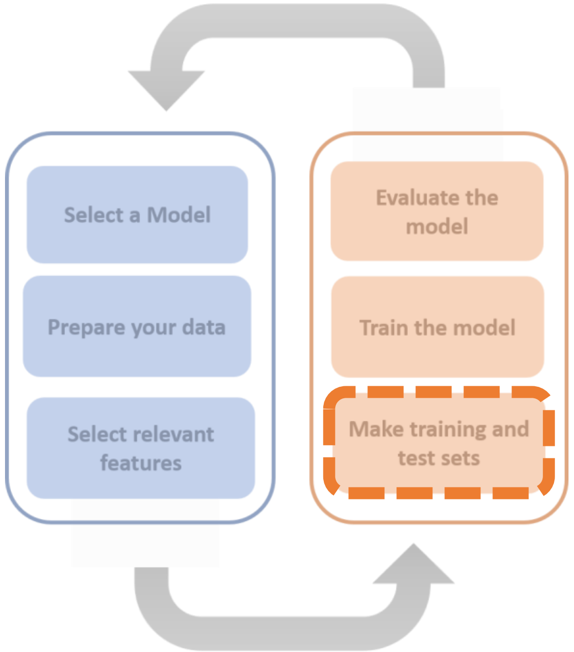
If your dataset is very high dimensional you often want to remove features. Reasons for this could be.....

- You might want to discard features where there are a lot of missing data or outlieres
- You might want to remove features that are highly correated with other features (reduce reduandancy)
- You might want to remove features that are of little relevance to the model
- You might not have enough datapoint for the dimensionality of your data (see curse of dimensionality ....)

In general it‘s not easy to know beforehand which features are relevant !

.... nevertheless, we will just make our live very easy now:

Area	Price
78 m^2	112.00 €
121 m^2	243.200 €
43 m^2	51.000 €
68 m^2	43.500 €
92 m^2	82.400 €



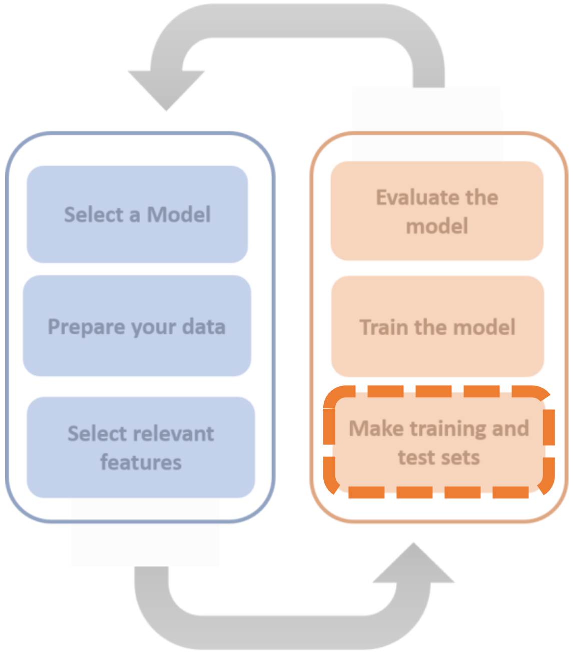
Make Train and Test Sets

….. this is easy….

```
In [ ]: from sklearn.model_selection import train_test_split
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

In [ ]:
```

….. But why should we do this ?



Make Train and Test Sets

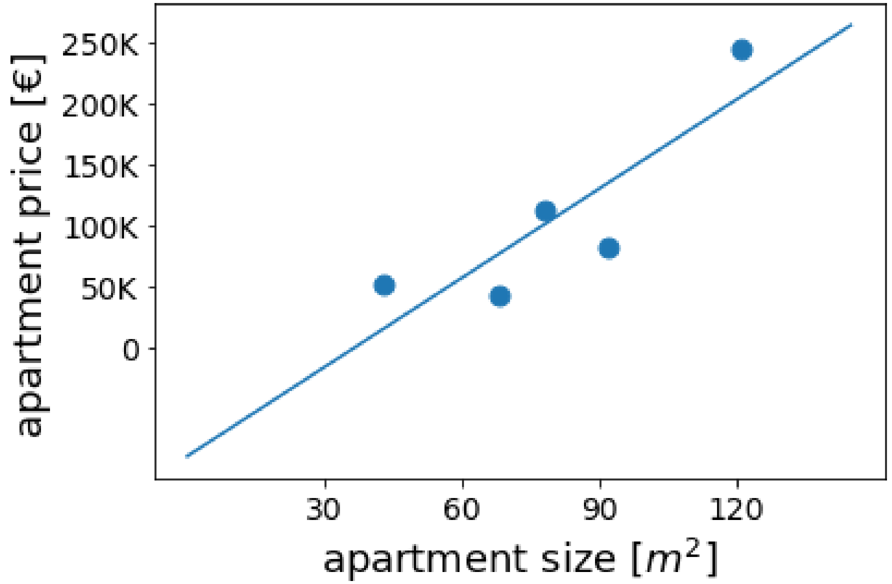
- During training the model learns to associate the datapoints with the labels.
- A useful (i.e generalizable) model will learn a meaningful association
- An unuseful (i.e not generalizable) model might just learn the association „by heart“
- There is no way of knowing if your model learned a meaningful association until you try what happens when it is applied to new data
- The test set is kept seperate during training so we can later prentend that these datapoints are new!

Example

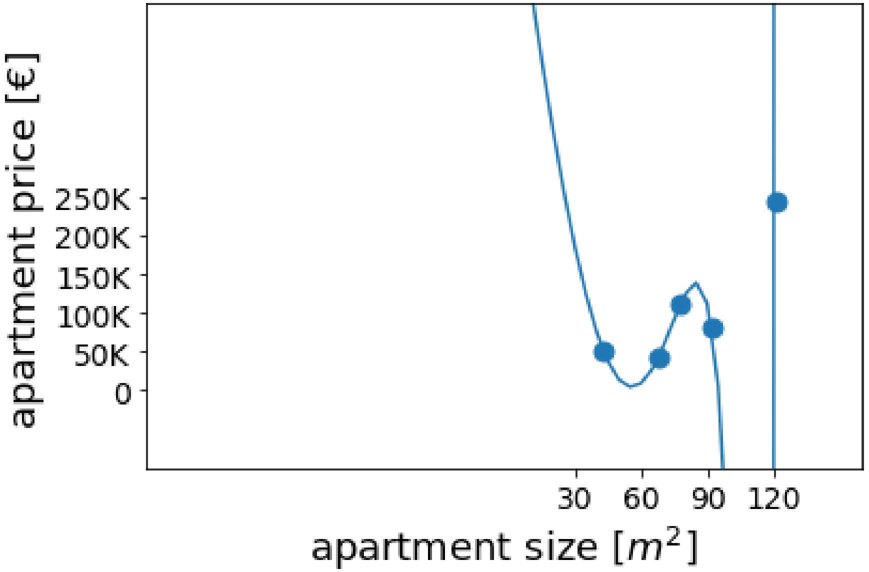
Make Train and Test Sets

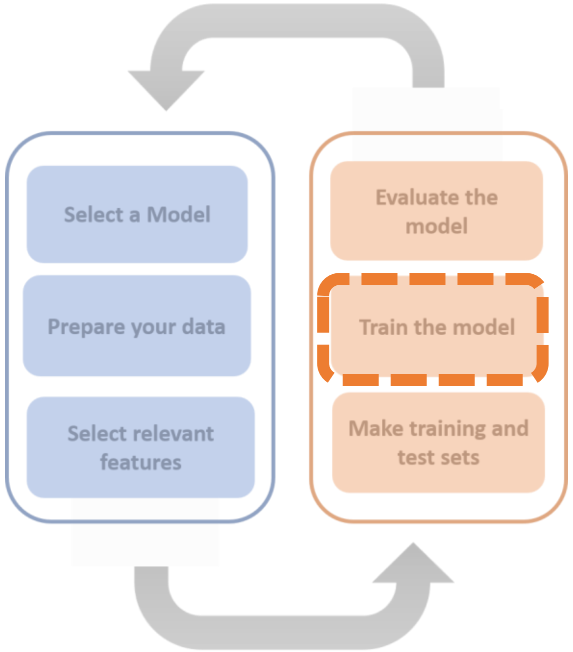
If the model performs very well on the training set but much worse on the test set, this is a good indicator that you are overfitting! In that case you should try to reduce the complecity of your model e.g. make your tree more shallow or decrease the degree of your polynomial in polynomial regression

OK fit of test data
Meaningful relationship



Amazing fit of test data
Meaningless relationship

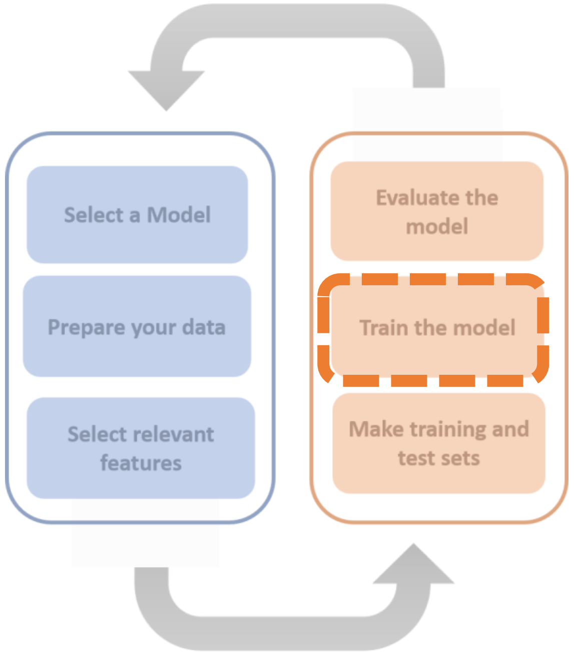




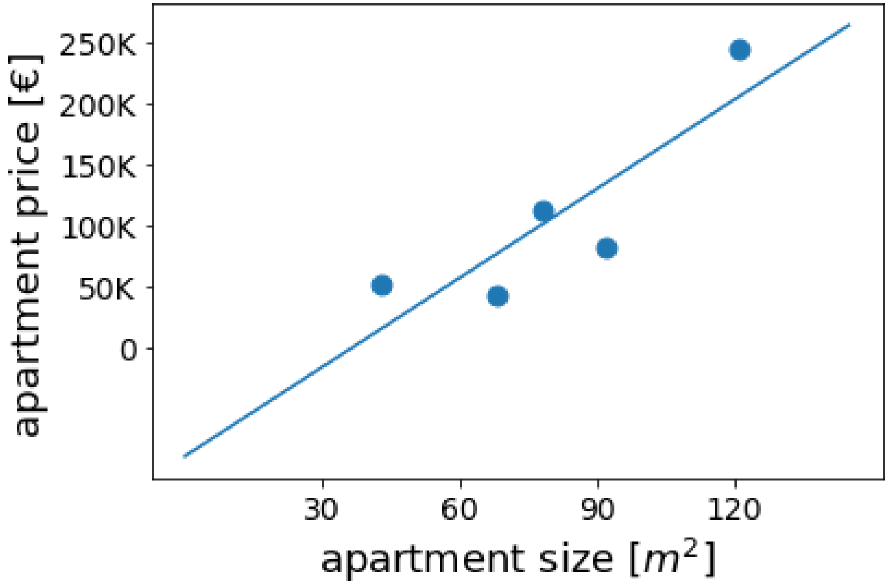
Train the model

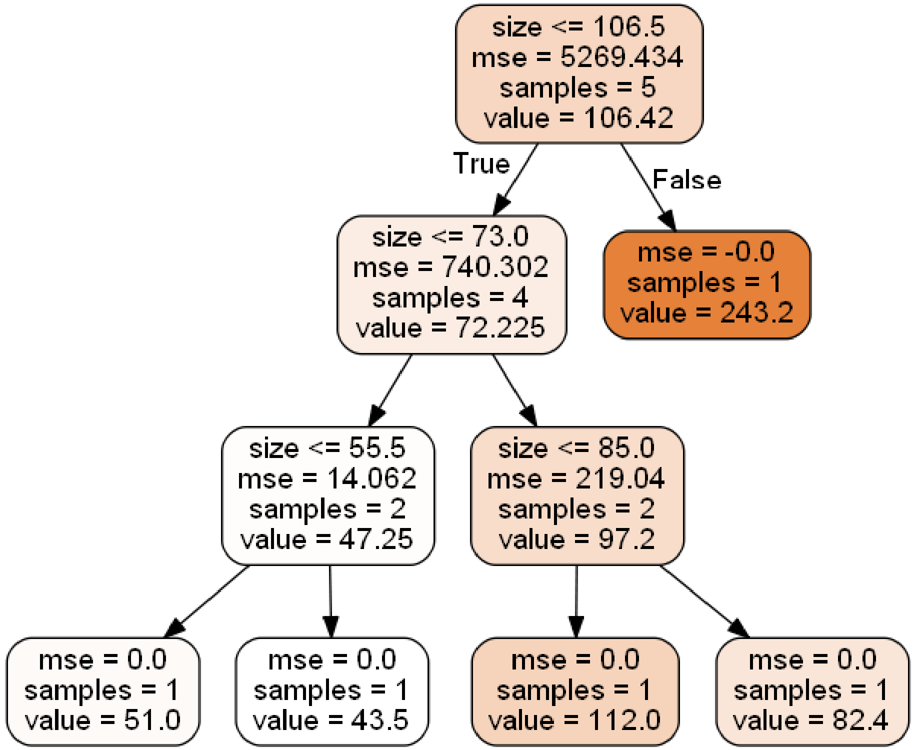
….. Also easy….

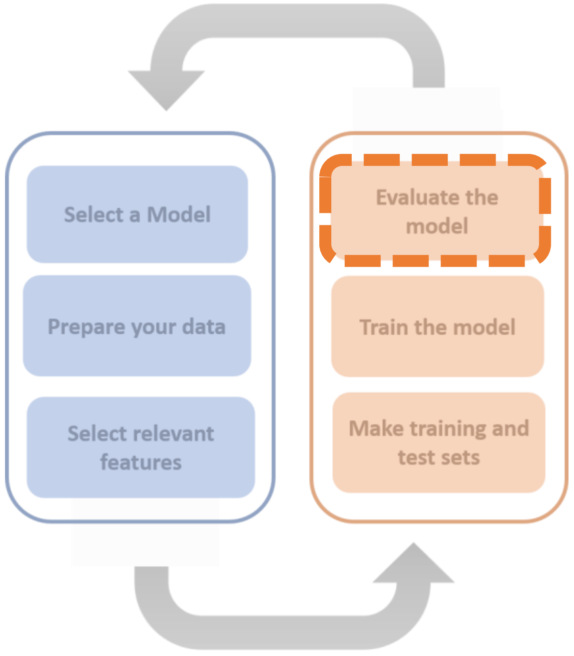
```
In [ ]: model.fit(X_train, y_train)
```



Train the model



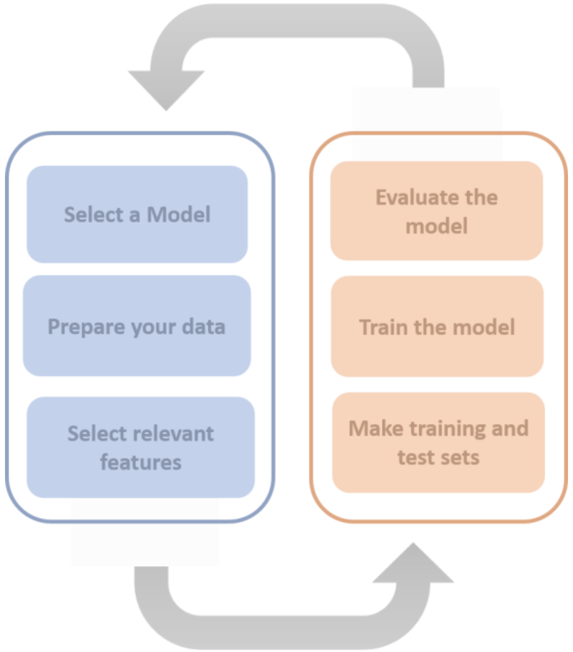




Evaluate the model

The model type determines what question you should ask…..

Classification	How many classes did I guess correctly?
Regression	How far is my predicted value from the true value?
Unsupervised Learning	Do the clusters seem reasonable?



Evaluate the model

Classification

Predicted lables	True Lables	Error ?
1	1	no
1	0	yes
0	1	yes
0	0	no
1	1	no

3 correct out of 5 → Accuracy = 3/5= 0.6

True positives (TP): II
True negatives (TN): I

False positives (FP): o
False negatives (FN): I

Precison: How many of those that I classified as 1 were actually 1s? $\frac{TP}{TP+FP} = \frac{2}{2+0} = 1$

Recall: How many of the 1s did I find? $\frac{TP}{TP+FN} = \frac{2}{2+1} = 0.67$

Example

Evaluate the model

Regression

What error measure you choose depends on the data and the intial question!

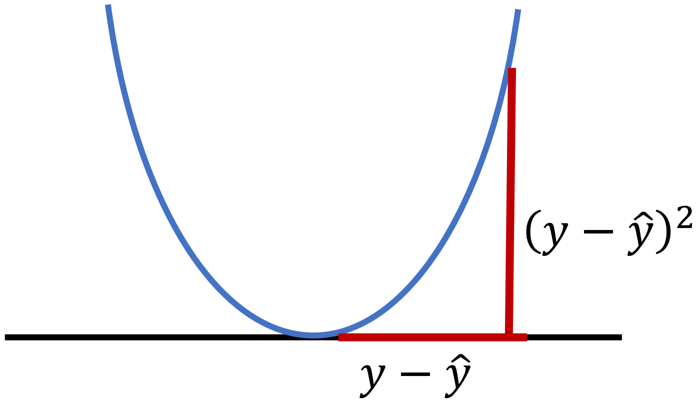
true Price	Predicted Price	MSE $(y - \hat{y})$	MAE $\lvert y - \hat{y} \rvert$
112.000 €	100.600 €	130.000.000	11.400 €
243.200 €	205.000€	1.457.000.000	38.1800 €
51.000 €	15.600€	1.253.000.000	35.400 €
43.500 €	76.300€	1.076.000.000	32.800 €
82.400 €	134.600€	2.723.000.000	52.200 €

Example

Evaluate the model

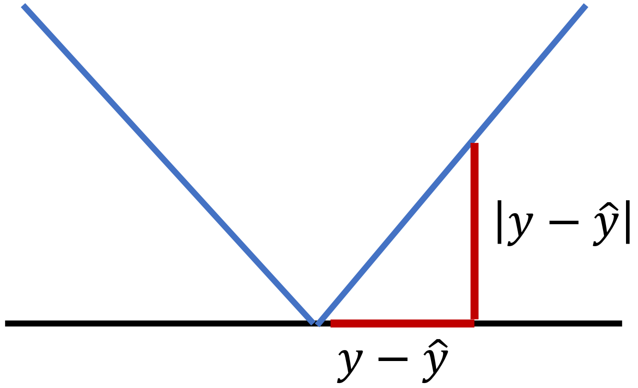
Regression

MSE: mean squared error

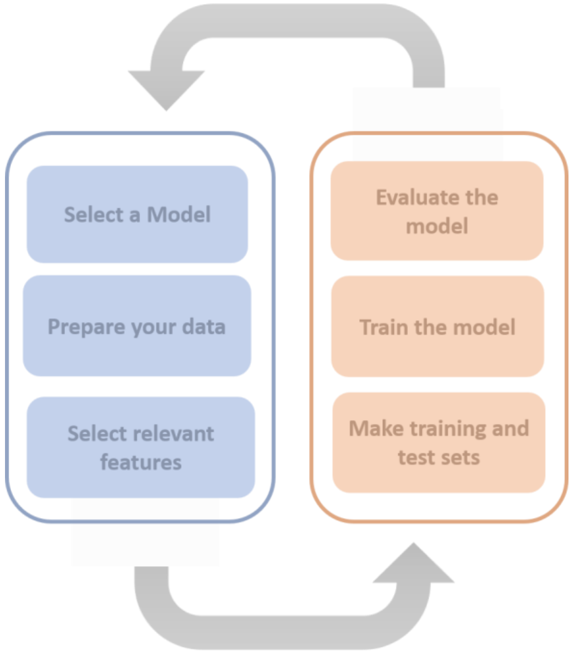


The squared error punishes small deviations less strongly and large deviations more strongly

ASE: mean absolute error



The absolute error punishes cares only how far you are from the true value

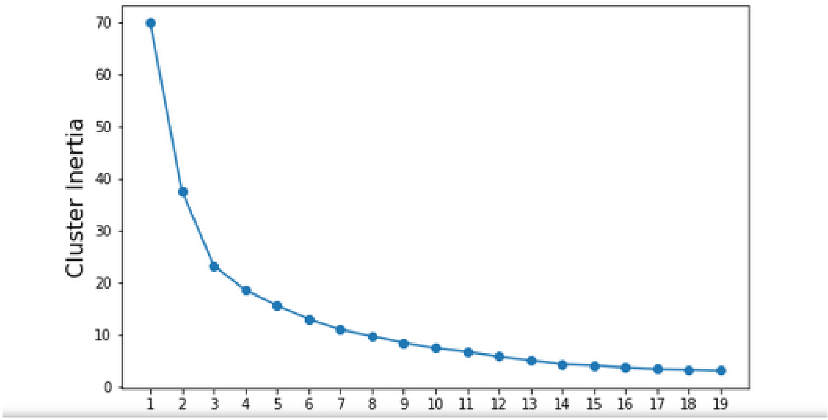


Evaluate the model

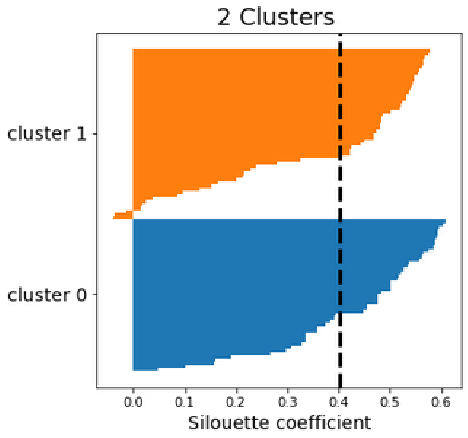
Clustering

In clustering there is no clear way to determine if you found the right clusters (except in cases where you‘re trying to cluster a labelled dataset).
But if you‘re using k-means clustering, ellbow plots and the silouette score can help you to get an idea if your clusters are reasonable

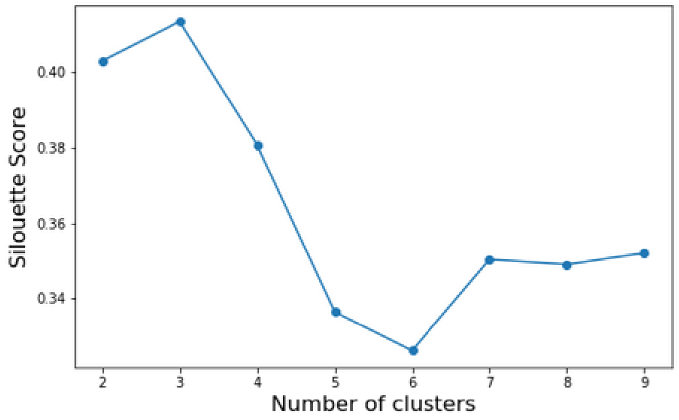
Ellbow plot

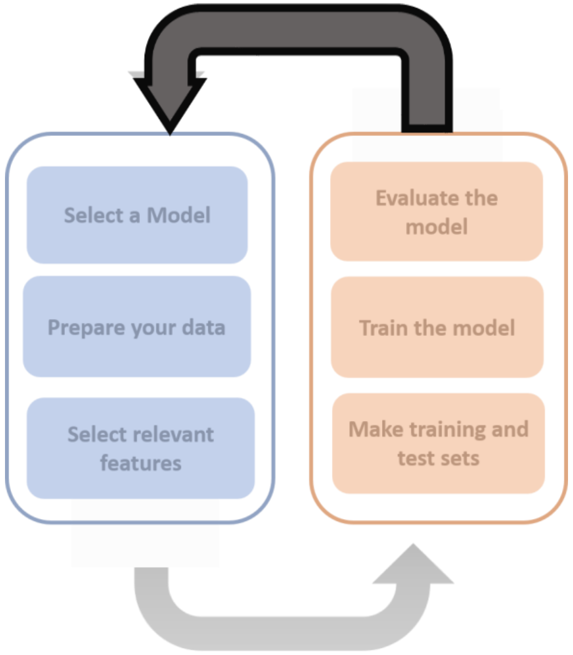


Silouette plot



Silouette Score





Back to square one

Use the the results of the model evaluation to ….

….. Select a different type of model
„turns out the problem is not linear, maybe use a tree instead of linear regression“

….. Select different parameters
„turns out my decison tree is overfitting, maybe use a random forrest“

….. Select different features
„the importantce score of my ensemble method tells me that 5 of my varaibles are not useful for prediction at all!“