Feature selection, scaling & engineering

The craftsmanship behind the magic

Feature selection. Keep the features with the most predictive value, drop the useless ones. Can be done manually or automatically.

Feature scaling. Bring features to the same range so that they have the same weight in the algorithm.

Feature engineering / extraction. Create new features, typically out of the current ones.

Feature selection

Why feature selection?

Simplicity	A good solution is a simple, lean solution. Keep the muscle, remove the fat!
Performance	Some models can just ‘ignore’ useless features (e.g. linear regression).. In others, noisy features have a negative impact (e.g. knn).
Noise	By chance, noise can adopt the shape of a pattern that your model will capture. This will cause bad predictions on new data.
Computational cost	Training models with large datasets can take a lot of time. You don’t want your computer to do useless calculations.

Low Variance

id	€	Height	Nationality
1	10	184	UK
2	20	176	UK
3	15	175	UK
4	5	169	UK

When (almost) all observations have the same value for a feature, that feature is not adding much. Remove it!

Use Variance Threshold from sklearn:
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html#sklearn.feature_selection.VarianceThreshold

Collinearity

If two features have a perfect correlation (r = 1), they are… just the same feature!

Just remove one of them

If two features have an almost perfect correlation, the presence of both is not adding much information.

Remove one or create a combination of both

What's the threshold between acceptable and 'too high' correlation?

It depends. If you can, try multiple strategies.

id	€	$	Right arm	Left arm	Height	Lacrosse?
1	10	11,77	75	75	184	True
2	20	23,53	70	69,9	176	False
3	15	17,65	71	71,02	175	True
4	5	5,88	66	66	169	False

How to drop features with high dimensionality: https://chrisalbon.com/machine_learning/feature_selection/drop_highly_correlated_features/

(Take some time to break the code apart and try to understand what's going on in there!)

Feature selection, the hacker way

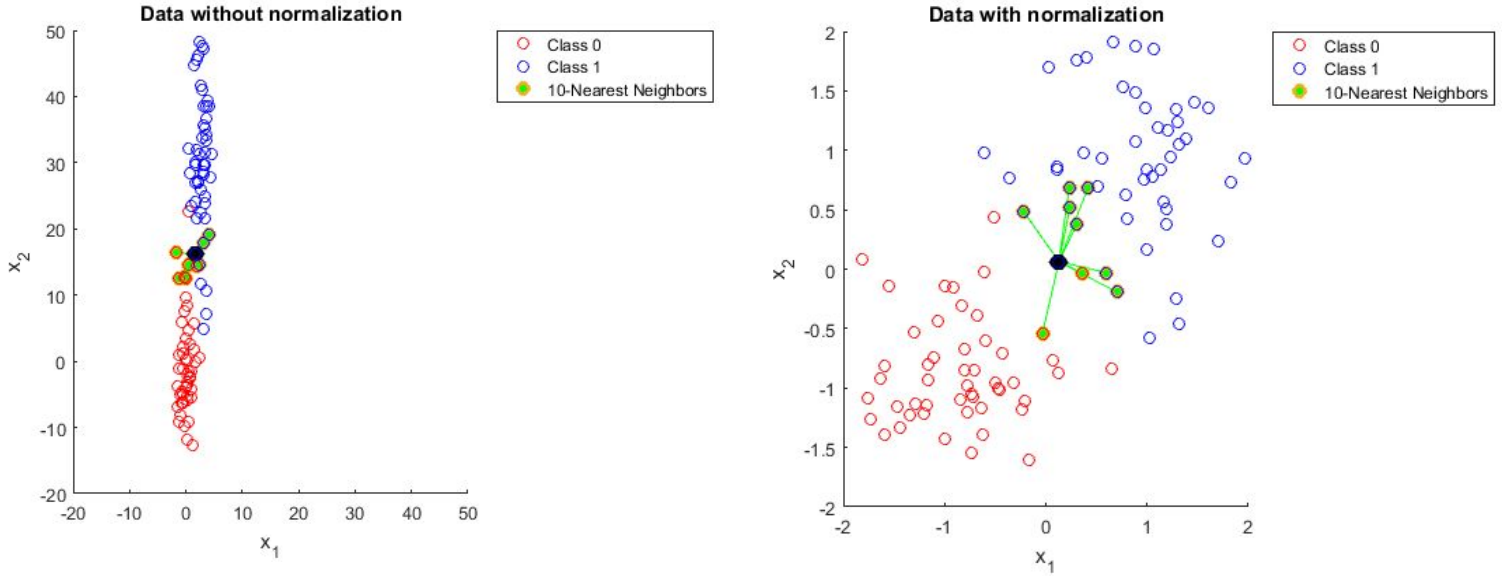
Use a transformer such as SelectKBest. It performs a statistical test to determine relationship between each variable and the target one. Then, selects the top K variables with the strongest relationships: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html#sklearn.feature_selection.SelectKBest

Some models (e.g. Random Forest) will give you the 'feature importance' after they have been fitted. This ranks the variables : https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html

Recursive Feature Elimination: fits a model many times, each time with a different combination of features, then ranks the features and takes the best ones (you define how many). https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html#sklearn.feature_selection.RFE

Feature scaling

Feature Scaling / Normalization



Scikit-Learn offers many types of scaling strategies, each one in a *transformer* from the submodule *preprocessing*. Mandatory reading:
https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02

Feature engineering / extraction

How would you encode geography… numerically?

id	€	Height	City
1	10	184	London
2	20	176	Liverpool
3	15	175	Bristol
4	5	169	Norwich

Coordinates?

Distance to the capital of the country / region?

Distance to the closest international airport / port?

Population / Density?

GDP per capita?

Get the region / country / continent & One Hot Encode?

Mentions on Twitter?

Feature engineering can be simple (i.e. merging width, height and depth of a product into a single variable called volume) or it can be creative & require data collection!

Bonus question: how would you encode time?