Train In Data

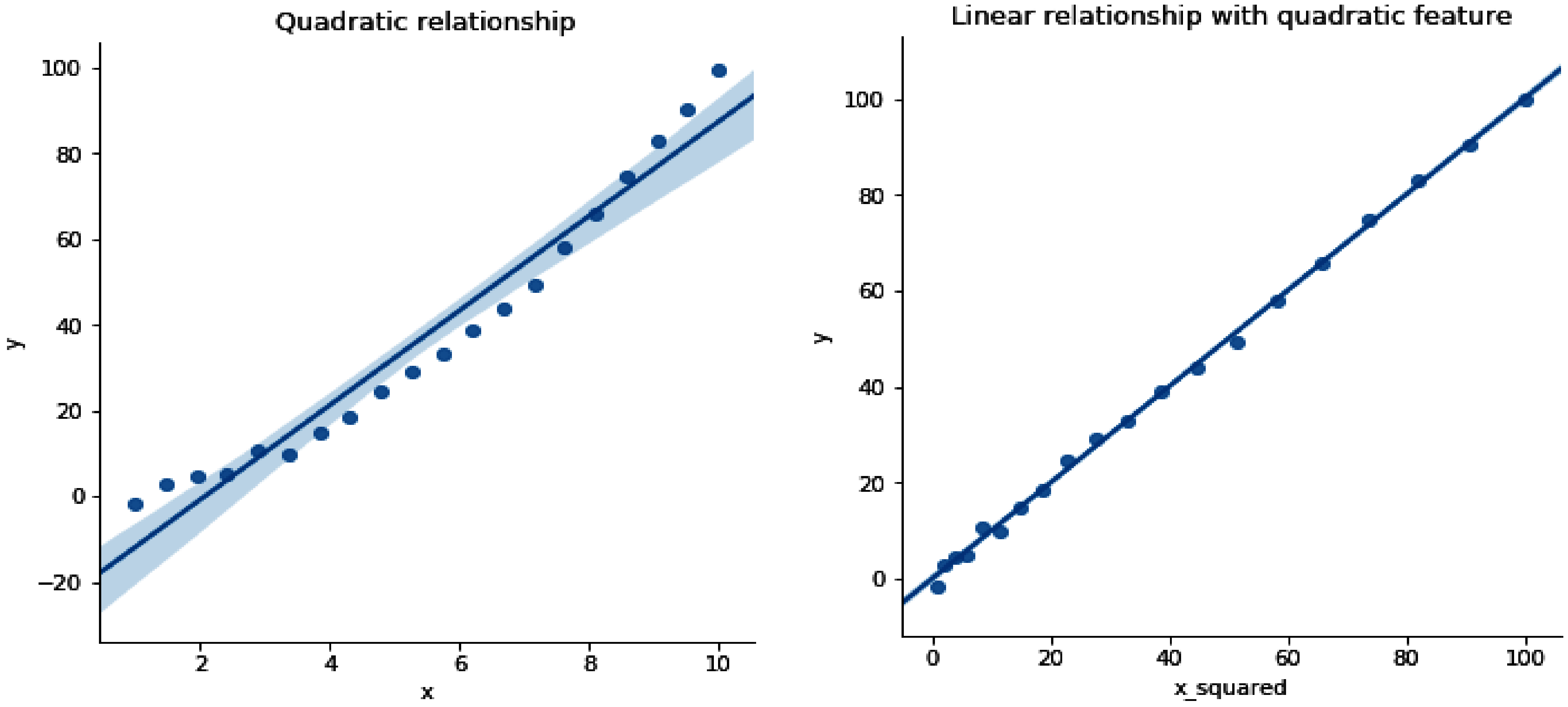
Polynomial Features

Polynomial features

Linear models assume a linear relationship between the predictors and the target.

Sometimes, the model is not linear to the feature, but it might be linear to a power of the feature.

Polynomial features



In this simulation, the target is linear to the square version of the original predictor.

Polynomial features

A polynomial feature is a new variable resulting from raising existing predictors to a certain exponent.

Typically, we use exponents up to 3 or 4. Otherwise, the functions are too flexible and lead to overfitting (Elements of Statistical Learning with applications in R)

Interactions

To capture feature interactions with linear models, it is common to add features resulting from the product of predictors.

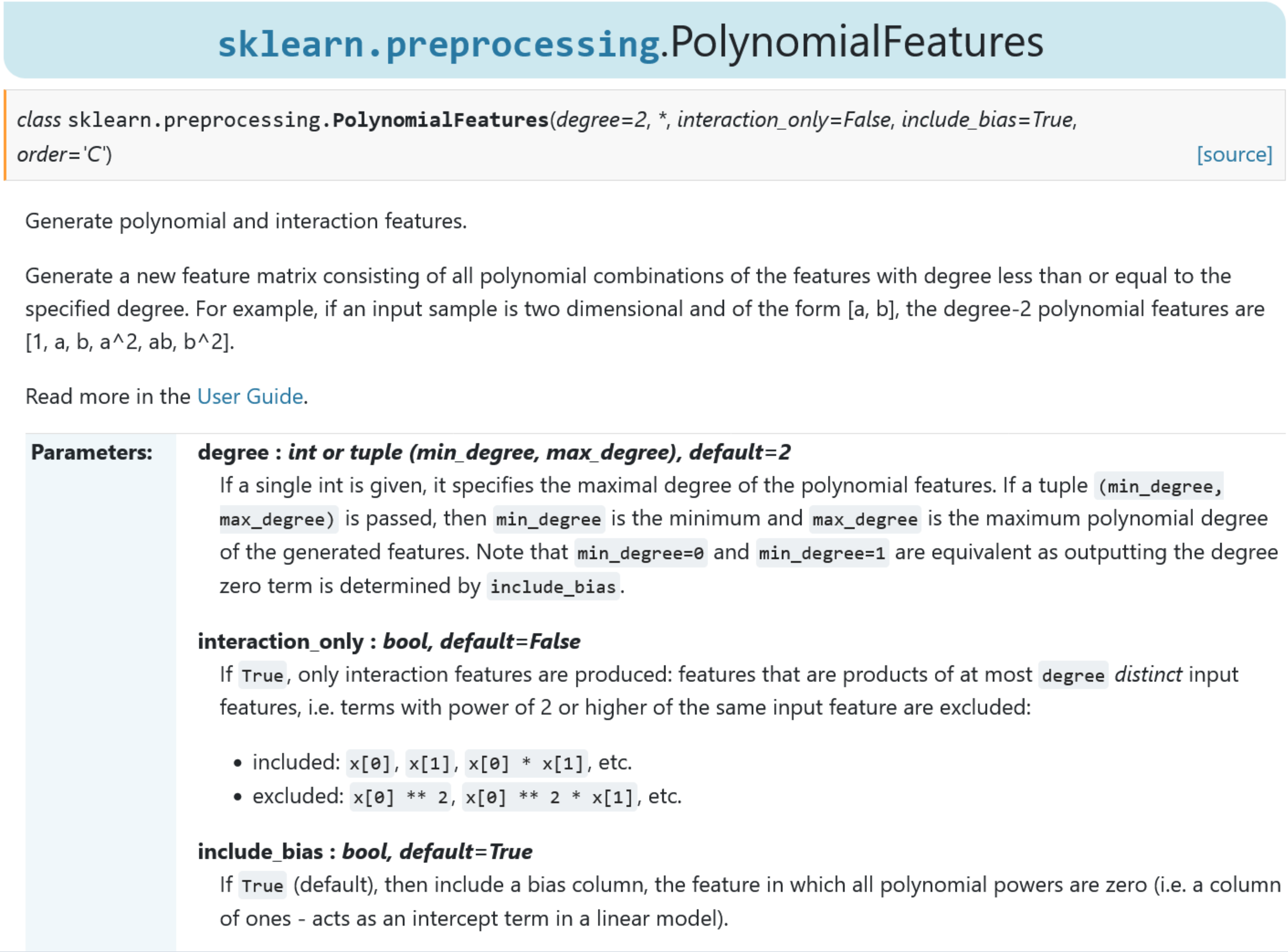
Examples that make sense: price x number of items bought

Examples that do not make sense: age x income

Polynomial expansion pros and cons

Plus: Can improve model performance.

Con: Can return features that are hard to understand or explain.



sklearn.preprocessing.PolynomialFeatures

class `sklearn.preprocessing.PolynomialFeatures`(*degree=2*, ***, *interaction_only=False*, *include_bias=True*, *order='C'*) [source]

Generate polynomial and interaction features.

Generate a new feature matrix consisting of all polynomial combinations of the features with degree less than or equal to the specified degree. For example, if an input sample is two dimensional and of the form [a, b], the degree-2 polynomial features are [1, a, b, a^2, ab, b^2].

Read more in the User Guide.

Parameters:

degree : *int or tuple (min_degree, max_degree), default=2*

If a single int is given, it specifies the maximal degree of the polynomial features. If a tuple `(min_degree, max_degree)` is passed, then `min_degree` is the minimum and `max_degree` is the maximum polynomial degree of the generated features. Note that `min_degree=0` and `min_degree=1` are equivalent as outputting the degree zero term is determined by `include_bias`.

interaction_only : *bool, default=False*

If `True`, only interaction features are produced: features that are products of at most `degree` *distinct* input features, i.e. terms with power of 2 or higher of the same input feature are excluded:

- included: `x[0]`, `x[1]`, `x[0] * x[1]`, etc.
- excluded: `x[0] ** 2`, `x[0] ** 2 * x[1]`, etc.

include_bias : *bool, default=True*

If `True` (default), then include a bias column, the feature in which all polynomial powers are zero (i.e. a column of ones - acts as an intercept term in a linear model).

Train In Data

THANK YOU

www.trainindata.com