Linear Regression - 3

Why perform feature scaling?

Ans: To bring all features to the same scale.
i.e. Removes ambiguity in feature importance

For example: if there are two features $f1,\ f2$:
- $f1$ value range >>> $f2$ value range → Weight of $f1$ >>> Weight of $f2$ ⇒ $f1$ important feature even though $f2$ is the important one.

Now Column Standardization makes: $f1$ value range ≈ $f2$ value range → Weight of $f1$ < Weight of $f2$ ⇒ $f2$ becomes important feature

What happens to R-squared if we add a new feature?

Ans: if the feature is relevant, R-square ↑.

But if the Feature is not relevant,
R-square should ↓ when model performance gets worse →

- The model's task is to minimize the loss
- So, if adding a feature is reducing the performance i.e. increasing the loss
- The model can simply assign small or zero weights to new features to avoid a decrease in performance.

Will the R-Square value increase or remain the same if we add a new feature?

Ans: Both are possible → small weights or zero weights can be assigned to new features which will keep performance the same.
model performance
The model can start making spurious associations with new features (i.e. overfit) causing model performance to increase on the train set.

As R-Square fails to compare performance and model complexity (no of features), what other metrics to use?

Ans: Adjusted R-Squared, defined as:

$$AdjR^2 = 1 - [\frac{(1-R^2)(n-1)}{(n-d-1)}],$$

where n is the number of samples, and d is the number of features

How does Adj R-Square compare performance and model complexity?

Ans: if the number of features (d) increases

with no significant feature:

- R^2 remains constant or slightly increased ⇒ (n-d-1) ↓ ⇒ Adj-R-squared ↓

with significant features:

- R^2 ↑ significantly ⇒ $[\frac{(1-R^2)(n-1)}{(n-d-1)}]$ ↓ ⇒ Adj-R-squared ↑