Feature Engineering: 6th lesson – Target Encoding

Most of the techniques we've seen in this course have been for numerical features. The technique we'll look at in this lesson, target encoding, is instead meant for categorical features. It's a method of encoding categories as numbers, like one-hot or label encoding, with the difference that it also uses the target to create the encoding. This makes it what we call a supervised feature engineering technique.

Target encoding:

A target encoding is any kind of encoding that replaces a feature's categories with some number derived from the target. A simple and effective version is to apply a group aggregation from Lesson 3, like the mean. Using the Automobiles dataset, this computes the average price of each vehicle's make.

autos["make\_encoded"] = autos.groupby("make")["price"].transform("mean")

autos[["make", "price", "make\_encoded"]].head(10)

make price make\_encoded

0 alfa-romero 13495 15498.333333

1 alfa-romero 16500 15498.333333

2 alfa-romero 16500 15498.333333

3 audi 13950 17859.166667

4 audi 17450 17859.166667

5 audi 15250 17859.166667

6 audi 17710 17859.166667

7 audi 18920 17859.166667

8 audi 23875 17859.166667

9 bmw 16430 26118.750000

This kind of target encoding is sometimes called a mean encoding. Applied to a binary target,it is also called bin counting (other names you might come across include: likelihood encoding, impact encoding, and leave-one-out encoding).

Smoothing:

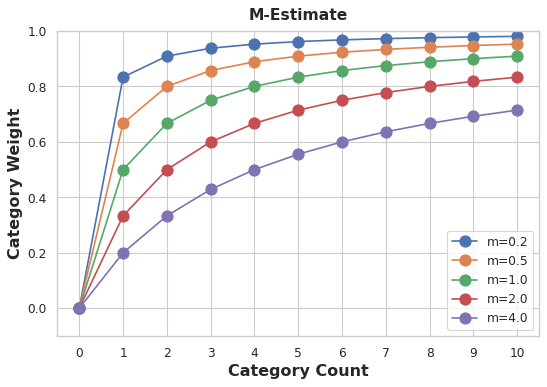
An encoding like this presents a couple of problems, however. First are unknown categories. Target encodings create a special risk of overfitting, which means they need to be trained on an independent "encoding" split. When you join the encoding to future splits, pandas will fill in missing values for any categories not present in the encoding split. These missing values you would have to impute somehow.

Second are *rare categories*. When a category only occurs a few times in the dataset, any statistics calculated on its group are unlikely to be very accurate. In the *Automobiles* dataset, the mercurcy make only occurs once. The "mean" price we calculated is just the price of that one vehicle, which might not be very representative of any Mercuries we might see in the future. Target encoding rare categories can make overfitting more likely.

A solution to these problems is to add smoothing. The idea is to blend the in-category average with the overall average. Rare categories get less weight on their category average, while missing categories just get the overall average.

In pseudocode: encoding = weight \* in\_category + (1 - weight) \* overall, where weightis a value between 0 and 1 calculated from the category frequency.

An easy way to determine the value for weight is to compute an m-estimate: weight = n / (n + m), where n is the total number of times that category occurs in the data. The parameterm determines the "smoothing factor". Larger values of m put more weight on the overall estimate.



In the *Automobiles* dataset, there are three cars with the make chevrolet. If you chose m=2.0,then the chevrolet category would be encoded with 60% of the average Chevrolet price plus 40% of the overall average price:

chevrolet = 0.6 \* 6000.00 + 0.4 \* 13285.03

When choosing a value for m, consider how noisy you expect the categories to be. Does the price of a vehicle vary a great deal within each make? Would you need a lot of data to get good estimates? If so, it could be better to choose a larger value for m; if the average price for each make were relatively stable, a smaller value could be okay.

Target encoding is great for:

* **High-cardinality features**

A feature with a large number of categories can be troublesome to encode: a one-hot encoding would generate too many features and alternatives, like a label encoding, might not be appropriate for that feature. A target encoding derives numbers for the categories using the feature's most important property: its relationship with the target.

* **Domain-motivated features**

From prior experience, you might suspect that a categorical feature should be important even if it scored poorly with a feature metric. A target encoding can help reveal a feature's true informativeness.