

Your model thinks blue is bigger than red.

D8A.ACADEMY/RESOURCES/ONEHOT-
VS-LABEL

THE BUG

red=0, blue=2	you invented an order
linear models	read blue as 2x green
KNN, k-means	read it as distance

THE THREE TOOLS

one-hot	a O/1 column per category
ordinal	integers, real order only
target encoding	category → mean of y

IN CODE

pd.get_dummies(df)	quick, notebooks
OneHotEncoder()	pipelines, prod
handle_unknown	← 'ignore' saves prod.

WHEN ORDER IS REAL

S < M < L < XL	ordinal is right here
OrdinalEncoder	you pass the order
ratings 1..5	already numeric, keep

TOO MANY CATEGORIES

10k zip codes	10k columns, do not
group the rare	tail → "other"
target encode + CV	← train folds only.

GOTCHAS

trees forgive it	splits ignore order
drop_first=True	for linear regression
fit on train	← or the test leaks in.

WORTH MEMORIZING

01 The bug and the fix, six lines

Same column, two encodings: one invents geometry, one does not.

```
# the bug: color has no order
df['color'] = df['color'].map(
    {'red': 0, 'green': 1, 'blue': 2})

# the fix: one column per color
X = pd.get_dummies(df, columns=['color'],
                    drop_first=True)
```

→ A linear model on the buggy version learns one coefficient for "colorness", so blue must pull exactly twice as hard as green. That constraint is pure fiction.

02 Trap: which encoder for which column

The 10-second decision, by column type and model family.

COLUMN	USE	BECAUSE
color, city, browser	one-hot	no order exists
S / M / L, satisfaction	ordinal, explicit order	the order is real
zip code, merchant id	target enc or "other"	one-hot explodes

→ Interview phrasing: trees can survive label encoding because splits only ask greater-or-less, but the moment a linear head or a distance metric appears, the fake order becomes signal.