ML PREPROCESSING: CATEGORICAL ENCODING

Matthew Hall

ONE HOT ENCODING

- When we want categorical variables in our regression / classification problems, we need to somehow represent them as integers

- However, we simply cannot label them all 1,2,3,4… as that would imply an incremental relationship

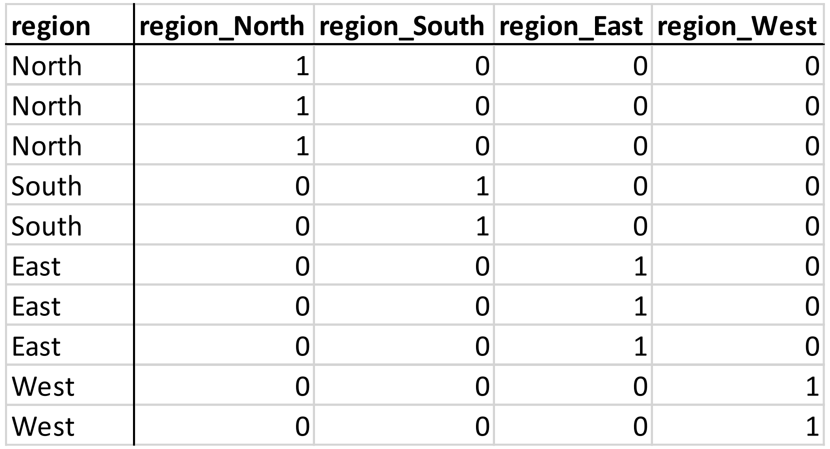
- To get around this, we often “One Hot” Encode The Categorical Variables

ONE HOT ENCODING EXAMPLE

- We Have A Column of North – South – East – West

- We Want To Account For This Effect in our Regression

- We Can Code For This Like On The Right:

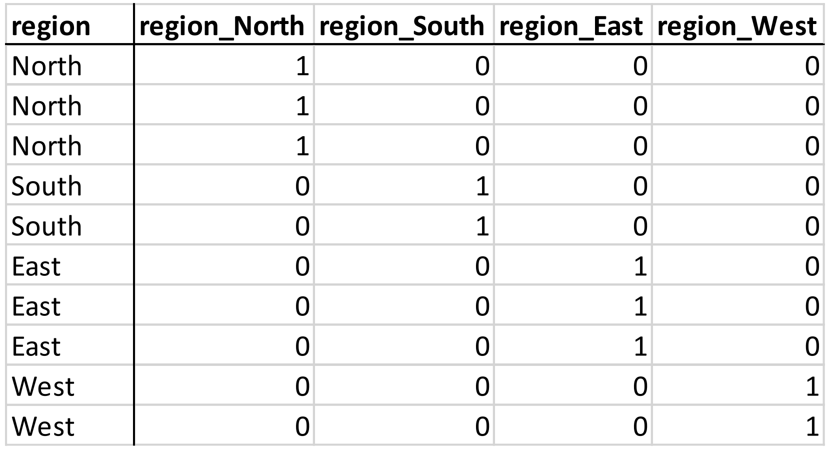


region	region_North	region_South	region_East	region_West
North	1	0	0	0
North	1	0	0	0
North	1	0	0	0
South	0	1	0	0
South	0	1	0	0
East	0	0	1	0
East	0	0	1	0
East	0	0	1	0
West	0	0	0	1
West	0	0	0	1

WORD OF CAUTION:

- **As it is on the right, we have collinearity, because *region_West* can be perfectly predicted from the first 3 columns**

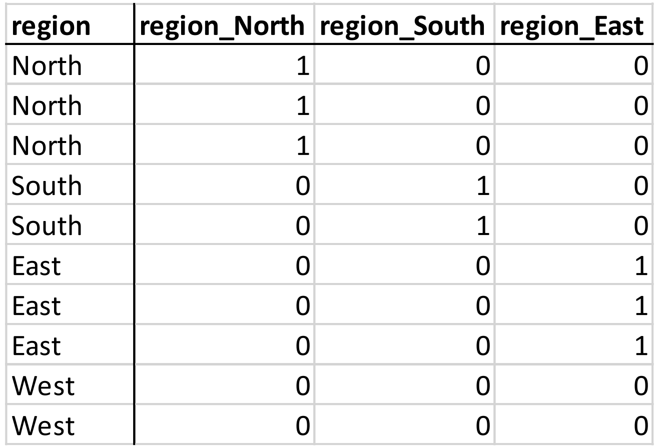
- **Excel, and Some OLS Regressions, Cannot Solve This Problem!**



region	region_North	region_South	region_East	region_West
North	1	0	0	0
North	1	0	0	0
North	1	0	0	0
South	0	1	0	0
South	0	1	0	0
East	0	0	1	0
East	0	0	1	0
East	0	0	1	0
West	0	0	0	1
West	0	0	0	1

FIXING THE COLINEARITY

- We Can Drop The Last Column to Fix This!
- *region_West* is simply all zeroes for the three other coefficients!



region	region_North	region_South	region_East
North	1	0	0
North	1	0	0
North	1	0	0
South	0	1	0
South	0	1	0
East	0	0	1
East	0	0	1
East	0	0	1
West	0	0	0
West	0	0	0

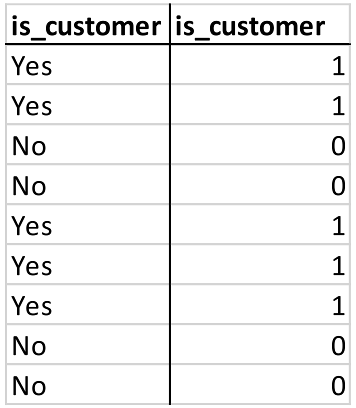
BINARY ENCODING

- If We Only Have Two Labels in a Column, We Binary Encode The Column.
- 0 and 1, with Each Corresponding to a Category
- Yes / No, True / False are Prime Candidates For Binary Encoding

BINARY ENCODING EXAMPLE

- 0 Usually Corresponds to False, and 1 True!

- This One Is Really Easy!



is_customer	is_customer
Yes	1
Yes	1
No	0
No	0
Yes	1
Yes	1
Yes	1
No	0
No	0

COLINEARITY & MODEL ROBUSTNESS

- **Earlier, We Dropped Off The First / Last Column To Prevent Collinearity From Appearing in the Model**

- **However, What Happens When We Want To Predict a Categorical Factor That Hasn’t Been Encoded?**

  - We Could Do All Zeroes, But That Would Correspond to the Dropped Off Category
  - If We Encode All Categories, All Zeroes Will Correspond to Unknowns in the Model

FIXING THE CONTRADICTION

- **We Can Use a Model Like Ridge Which Can Handle Colinear Variables**
  - However, We Lose Interpretation on The Coefficients
  - It’s Still Usable for Predictions!
- **We Can Encode Unknowns as the Same as the Dropped Off Category**
  - Introduces some bias, but if the unknowns are rare, this should be fine!
- **There’s No Magical Solution, However, We Don’t Want Our Models to Break!**

WHY IN PYTHON

- We Can Use SKLearn to Process This In The Backend, And Not Directly Encode Everything
- Prevent Highly Tedious Work in Excel – Imagine Encoding 80 Factors to Be Zero-One!
- And What Happens If It's Subject to Variability?

ASSOCIATED NOTEBOOK

- ***mod10-preprocessing-categorical-encoding.ipynb***
- **Let’s Get Into It!**

MELIORA