Categorical Variables

In this reading, you will review how and why categorical variables must be handled differently than continuous variables in multiple linear regression analysis. You will review the concept and process of one hot encoding and an example regression problem and sample dataset.

Interpreting results is a critical part of the PACE workflow (PACE: plan, analyze, construct, execute) and sharing insights with stakeholders. Interpret categorical variables in the following manner: when a smoker changes to a nonsmoker, the dependent variable increases/decreases by the categorical coefficient of 'x.'

One hot encoding

One hot encoding technique is a data transformation technique that turns categorical variables into binary ones. With one-hot encoding, you need as many dummy variables as there are unique categorical labels minus one dummy variable to reduce the effects of multicollinearity. For example, a categorical variable with high, medium, and low labels would add two new binary (0/1) variables to represent the medium and low categories. A variable with poor, good, better, and best labels adds three new binary variables per default, one hot encoding setting.

Use the following three Python libraries to help perform one hot encoding. Each library has more or fewer features. For example, sometimes you don't want to drop the parent variable, and the sklearn library OneHotEncoder() function includes this feature while the pandas .get_dummies() function does not. The statsmodel package automatically drops the parent variable and a dummy label when there are more than two categorical labels.

Option A: Pandas Library

Pandas .get_dummies(data, drop_first=False) function automates categorical variable creation. By default it drops the parent categorical variable. Set the drop_first=True when you want to exclude one of the children binary variables created. In the following example, the prefix=() parameter

allows you to add a shortname to the title of new categorical variables to help with their coefficient interpretation.

```
import pandas as pd
df1 = pd.get_dummies(df0, prefix=['salary','Department'],
                     columns = ['salary','Department'] , drop_first=True)
```

In the event you separate the categorical variables, or need to merge another variable from another dataframe, use the pandas concatenate function (pd.concat()) to combine the data.

```
df1 = pd.concat([df0,dummies],axis='columns')
```

Option B: Statsmodels library

When using the statsmodels package, add the character C around a categorical variable, such as C(genre) in step C below, to have it encode a categorical variable containing labels. Statsmodel automatically drops the parent variable and the first binary label variable.

1.
```
# Import ols function
from statsmodels.formula.api import ols
```

2.
```
# Set to df or subset Data
ols_data = df_video_data
```

3.
```
# Write out formula
ols_formula = "number_of_streams ~ budget + C(genre)"
```

4.
```
# Build OLS, fit model to data, review model summary
OLS = ols(formula = ols_formula, data = ols_data)
model = OLS.fit()
model.summary()
```

Option C: SkLearn Library

The scikit learn library has the most specific one hot encoding parameters. Use the following summary to help make your choices and ensure to bookmark OneHotEncoders(<parameters::>) for assistance later on.

- # drop = None, keep all binary variables created
- # drop = 'first', drop the first categorical binary variable
- # drop = 'if_binary', drops one variable when a parent categorical variable has only 2 labels
- # sparse = set to true when the default of False is not generating data. This usually happens when the data set has a lot of zeros. You need to set sparse = True so sklearn processes it correctly.
- # dtype = # Desired data type of the output, such as int, float 64
- # handle_unknown = 'error' or 'ignore' automates viewing or ignore exceptions

1. # import the library
   # ohe = OneHotEncoder(drop=,spare =T/F,dtype =<int/float>,handle_unknown='')

   ```
   from sklearn.preprocessing import OneHotEncoder
   ohe = OneHotEncoder(drop='first',sparse=False,dtype = float64,
   handle_unknown='ignore')
   ```

2. #combine sklearns fit_transform() with ohe to transform the new child binary
   #variables into a column format to help easily combine into existing dataframe

   ```
   ohe.fit_transform(raw_df['salary'].unique().reshape(-1, 1))
   transformed_data = ohe.transform(raw_df['salary'].to_numpy().reshape(-1, 1))
   ```

3. # combine original dataframe (raw_df) with new binary variable dataframe
   # drop the parent variable, for example 'salary' on the column axis (=1).

   ```
   ohe_df = pd.concat([raw_df, transformed_data], axis=1).drop(['salary'], axis=1) # axis=0
   <row>, axis=1 <column>
   ohe_df.head()
   ```

Why do we need to handle categorical variables differently than continuous variables?

Regression analysis relies on statistics, which relies on mathematical principles. But, a lot of really interesting data is not inherently numerical. For example, product type, clothing size, country of origin, etc. But, we would lose a lot of information if we weren't able to use any categorical variables. By processing categorical variables into numerical data, you can allow the model to learn from categorical data, and thereby gaining richer insights.

Example regression problem

Let's say that you're working at a streaming service, and are trying to determine what kinds of movies should be added to the offerings on the site. One way to approach this problem is to gather data about movies that are already on your platform. You can use the data to determine what factors are correlated with high streaming numbers. For example, a data sample may be something like this:

Movie ID	Genre	Duration (minutes)	Days on streaming platform	Budget ($)	MPAA rating	Rating	Number of streams
movie_001	Action	137	14	215,000,000	R	5	11,349
movie_002	Children's	89	35	85,000,000	G	3	8,124
movie_003	Romance	126	50	120,000,000	PG-13	2	64,871
movie_004	Adventure	102	157	180,000,000	PG	4	382,589
movie_005	Comedy	148	89	50,000,000	R	3	225,172

You have 8 variables in question. There are categorical variables and continuous variables. It's clear that duration, days on streaming platform, budget, and number of streams are continuous. Additionally, genre and MPAA rating are categorical variables.

The regression model

Now that you have established the variables at our disposal, you can write out a linear regression model for the purposes of our business problem. Let's say that the dependent variable (y) is number of streams, and the independent variables are duration, days on streaming platform, budget, revenue, and MPAA rating. So the formula would be:

$$Streams = \beta_0 X_0 + \beta_{duration} X_{duration} + \beta_{days} X_{days} + \beta_{budget} X_{budget} + \beta_{revenue} X_{revenue} + \beta_{MPAA} X_{MPAA}$$

But, we need to one hot encode the MPAA rating since it is categorical.

One hot encoding MPAA rating

Based on the data, and the MPAA rating system, you know that there are 5 categories:

- G (general audiences)
- PG (parental guidance suggested)
- PG-13 (parents strongly cautioned)
- R (restricted)
- NC-17 (no one 17 and under admitted)

Since there are 5 categories, then we need 4 binary variables to capture the same amount of information. Let's say that the G-rating is the baseline, so we will exclude a $\beta_G X_G$ term from the equation, and interpreting the beta coefficients is in relation to G-rated movies.

$$\beta_{MPAA} X_{MPAA} \rightarrow \beta_{PG} X_{PG} + \beta_{PG-13} X_{PG-13} + \beta_R X_R + \beta_{NC-17} X_{NC-17}$$

In this case, the beta coefficients represent the following:

- Holding all other variables constant, β_{PG} is the difference in the number of streams between a G-rated and PG-rated movie.
- Holding all other variables constant, β_{PG-13} is the difference in the number of streams between a G-rated and a PG-13 movie.
- Holding all other variables constant, β_R is the difference in the number of streams between a G-rated and an R-rated movie.

- Holding all other variables constant, β_{NC-17} is the difference in the number of streams between a G-rated and an NC-17 movie.

These kinds of statements will help you to report multiple regression results to stakeholders in an interpretable way.

Seemingly numerical variables

But what about the rating variable? The observed data seems numerical; we have ratings of 1, 2, 3, 4, or 5. But these kinds of ratings variables—whether out of 3, 5, 10, or another value—are actually categorical. This is because there is no set difference between a 3-star movie and a 4-star movie. Every person that rates a movie thinks about the rating system a bit differently. So 1-star movies are a category; 2-star movies are another category. In this particular case, the categories have a rank to them: 5-star movies are better than 4-star movies, and so on. There are other examples of variables that seem continuous, but are actually categorical. For example:

- Voting district
- Zip code
- Area code
- Floor number

Keep these examples in mind as you explore data sets in the future. Be alert for seemingly numerical variables in your data sets that are actually categorical!

Key takeaways

- One hot encoding is a data transformation technique for handling categorical data.
- One hot encoding helps the model quantify the variable relationships with categorical data.
- Sometimes numerical data seems continuous but is actually categorical.