**The cutoff point**

In this exercise, and throughout this chapter, you'll be working with the restaurants DataFrame which has data on various restaurants. Your ultimate goal is to create a restaurant recommendation engine, but you need to first clean your data.

This version of restaurants has been collected from many sources, where the cuisine\_type column is riddled with typos, and should contain only italian, american and asian cuisine types. There are so many unique categories that remapping them manually isn't scalable, and it's best to use string similarity instead.

Before doing so, you want to establish the cutoff point for the similarity score using the fuzzywuzzy's process.extract() function by finding the similarity score of the most *distant* typo of each category.

* Import process from fuzzywuzzy.
* Store the unique cuisine\_types into unique\_types.
* Calculate the similarity of 'asian', 'american', and 'italian' to all possible cuisine\_types using process.extract(), while returning all possible matches.
* # Import process from fuzzywuzzy
* from fuzzywuzzy import process
* # Store the unique values of cuisine\_type in unique\_types
* unique\_types = restaurants['cuisine\_type'].unique()
* # Calculate similarity of 'asian' to all values of unique\_types
* print(process.extract('asian', unique\_types, limit = len(unique\_types)))
* # Calculate similarity of 'american' to all values of unique\_types
* print(process.extract('american', unique\_types, limit = len(unique\_types)))
* # Calculate similarity of 'italian' to all values of unique\_types
* print(process.extract('italian', unique\_types, limit = len(unique\_types)))

# Remapping categories II

In the last exercise, you determined that the distance cutoff point for remapping typos of 'american', 'asian', and 'italian' cuisine types stored in the cuisine\_type column should be 80.

In this exercise, you're going to put it all together by finding matches with similarity scores equal to or higher than 80 by using fuzywuzzy.process's extract() function, for each correct cuisine type, and replacing these matches with it. Remember, when comparing a string with an array of strings using process.extract(), the output is a list of tuples where each is formatted like:

(closest match, similarity score, index of match)

The restaurants DataFrame is in your environment, and you have access to a categories list containing the correct cuisine types ('italian', 'asian', and 'american').

* Return all of the unique values in the cuisine\_type column of restaurants.
* # Inspect the unique values of the cuisine\_type column
* print(restaurants['cuisine\_type'].unique())