

Your Turn: The Challenge

Transform the features in the cleaned Keiko customers dataset into the format required for fine-tuning.

Setup

You'll want to use the *transformed and cleaned* version of the Keiko customers dataset you exported at the end the last phase of the course project.

Alternatively, you can use *my* version of the same dataset, which you can find in the course resources.

Requirements

1. **Write a function called `row_to_text()`** that takes a single row of the DataFrame as input and returns a text description of that customer as a string. The description must include these 4 fields, each labeled clearly:
 - tenure
 - contract
 - monthly_charges
 - internet_service
2. **Create a new text column** in the DataFrame by apply-ing `row_to_text()` to every row.

Example Output

For a single row, your text value should look something like this:

Tenure: 2 months. Contract: Month-to-month. Monthly charges: \$89.50. Internet service: Fiber optic.

Hints

To keep things straightforward, try:

1. Creating a text string “sentence” pertaining to one of the 7 fields
2. Assigning this text string to a variable
3. Concatenating the remaining “sentences” pertaining to the remaining 6 fields to the same variable, one at a time. (Make sure to include spaces between the sentences!)

Also, since you’ll be concatenating values from numeric fields like “monthly charges” to the giant text string you’re building out, you’ll need to *convert* those values to a string data type. You *could* do this piecemeal with a conversion function like `str`, but a much simpler approach is to leverage something called an *f-string*.

If you’re not familiar, f-strings are a Python feature that let you drop variables and expressions - **regardless** of their data type - directly into a string by prefixing it with the letter “f” (hence, f-strings!) and wrapping the inserted pieces in curly braces `{}`. This lets you build out complex text strings without manually converting types or stitching things together with `+`.

Below is an example of f-string syntax (note the “f” at the beginning of the expression, *before* the quotes!):

```
age = 7

sentence = f"My dog is {age} years old, and {age * 7} years old
in human years."

print(sentence)
```

Output: My dog is 7 years old, and 49 years old in human years.