# Parsing Examples

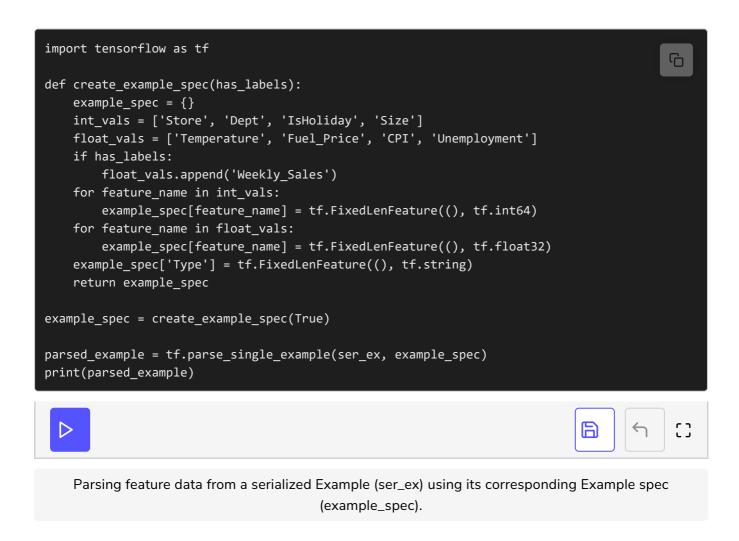
Set up a function for parsing Example objects with and without labels.

#### **Chapter Goals:**

• Create a function to parse feature data from serialized Example objects

### A. Extracting feature data

The input pipeline consists of reading serialized Example objects from the TFRecords file and parsing feature data from the serialized Examples. We parse feature data for a single Example (which represents data for one DataFrame row) using the tf.parse\_single\_example function.



#### B. Labeled data

For both training and evaluation, we require the data to be labeled in order to

calculate the loss for our machine learning model. Since our model is trained

to predict weekly sales, we use the 'Weekly\_Sales' feature as the label for each data observation.

This means that every feature apart from 'Weekly\_Sales' is used as input for the model, which then predicts a 'Weekly\_Sales' value based on these input features. The value is compared to the actual 'Weekly\_Sales' label to assess the model's performance.

## Time to Code!

In this chapter you'll be completing the parse\_feature function, which parses
features from a serialized Example object.

We first use the input Example spec argument, <a href="mailto:example\_spec">example\_spec</a>, to parse a dictionary of features from the input serialized Example, <a href="mailto:ser\_ex">ser\_ex</a>.

Set parsed\_features equal to tf.parse\_single\_example with ser\_ex and example\_spec as the required arguments.

The 'Weekly\_Sales' feature is not actually used as an input for the machine learning model. Instead, it is used as a label during training and evaluation.

Therefore, we'll separate the 'Weekly\_Sales' feature from the rest of the features (if it's even present).

Create a dictionary called features which is identical to parsed\_features, but does not contain the 'Weekly\_Sales' key.

If has\_labels is False (meaning the 'Weekly\_Sales' feature is not present), we'll just return the rest of the features.

Create an if block that checks if has\_labels is False. Inside the if block, return features.

If has\_labels is True, we can extract the 'Weekly\_Sales' feature from the parsed features, which is used as a label.

Outside the if block, set label equal to the value in the parsed\_features dictionary corresponding to 'Weekly\_Sales'. Return the tuple features, label.

```
import tensorflow as tf

# Helper function to convert serialized Example objects into features
def parse_features(ser_ex, example_spec, has_labels):
    # CODE HERE
    pass
```









