

# 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.

```
import tensorflow as tf

def create_example_spec(has_labels):
    example_spec = {}
    int_vals = ['Store', 'Dept', 'IsHoliday', 'Size']
    float_vals = ['Temperature', 'Fuel_Price', 'CPI', 'Unemployment']
    if has_labels:
        float_vals.append('Weekly_Sales')
    for feature_name in int_vals:
        example_spec[feature_name] = tf.FixedLenFeature([], tf.int64)
    for feature_name in float_vals:
        example_spec[feature_name] = tf.FixedLenFeature([], tf.float32)
    example_spec['Type'] = tf.FixedLenFeature([], tf.string)
    return example_spec

example_spec = create_example_spec(True)

parsed_example = tf.parse_single_example(serialized_example, example_spec)
print(parsed_example)
```



Parsing feature data from a serialized Example (ser\_ex) using its corresponding Example spec (example\_spec).

## 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, `example_spec`, to parse a dictionary of features from the input serialized Example, `ser_ex`.

**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(serialized_example, example_spec, has_labels):
    # CODE HERE
    pass
```

