Example Spec

Create an Example spec used for parsing serialized Examples.

Chapter Goals:

- Create the Example spec that's used to parse serialized Example objects

A. Creating the Example spec

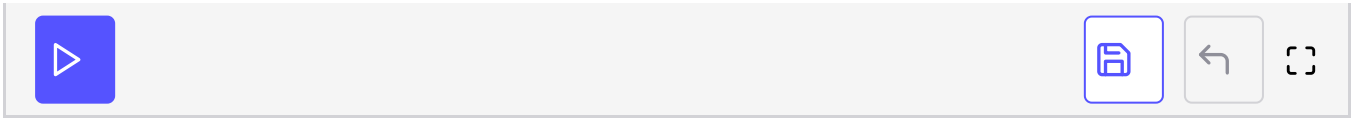
Since the data is now stored as serialized Example objects in TFRecords files, we need to create an Example spec, which allows us to parse the serialized Examples in the input pipeline. The Example spec gives specifications on each of the dataset's features, specifically the shape and type of the feature's values.

The Example spec is just a Python dictionary, mapping feature names to `FixedLenFeature` objects. For our dataset, each of the `FixedLenFeature` objects has shape `()`. This is because each feature contains a single value per data observation, i.e. each feature of our dataset's Example objects contains one value (as opposed to an array of values).

The type of each `FixedLenFeature` depends on the type of values in the feature. Integer valued features will have `tf.int64` type, float valued features will have `tf.float32` type, and string valued features (in our case, only the `'Type'` feature) will have `tf.string` type.

```
import tensorflow as tf

example_spec = {}
example_spec['Store'] = tf.FixedLenFeature((), tf.int64)
example_spec['CPI'] = tf.FixedLenFeature((), tf.float32)
example_spec['Type'] = tf.FixedLenFeature((), tf.string)
```



Creating the Example spec for our dataset. The above code's example_spec, which is incomplete, contains specifications for the 'Store', 'CPI', and 'Type' features.

Time to Code!

In this chapter you'll be completing the `create_example_spec` function, which creates an Example spec based on the dataset features. The function is already filled with code that provides the feature names for the dataset.

For each of the integer features, we'll create a `FixedLenFeature` object with shape `()` and type `tf.int64`.

Create a `for` loop that iterates through `int_vals` using a variable named `feature_name`.
Inside the `for` loop, set `feature_name` as a key in `example_spec`. The value it maps to will be `tf.FixedLenFeature` initialized with the specified shape and type.

We'll now repeat the previous step for each of the float features. The shape of the `FixedLenFeature` objects is still `()`, but now the type is `tf.float32`.

Create another `for` loop (outside the previous one) that iterates through `float_vals` using a variable named `feature_name`.
Inside the `for` loop, set `feature_name` as a key in `example_spec`. The value it maps to will be `tf.FixedLenFeature` initialized with the specified shape and type.

Finally, we'll create a `FixedLenFeature` for the only string feature (`'Type'`), which will have shape `()` and type `tf.string`.

Outside the `for` loops, set `example_spec['Type']` to a `tf.FixedLenFeature` object initialized with the specified shape and type. Then return `example_spec`.

```
import tensorflow as tf

# Create the spec used when parsing the Example object
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')
    # CODE HERE
```