Integer Features

Learn about the integer features used in the dataset.

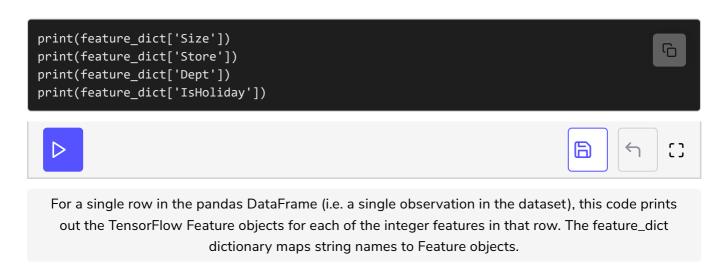
Chapter Goals:

• Add the integer features of a DataFrame's row to a feature dictionary

A. Using Example objects

Each row of the final pandas DataFrame from the Data Analysis Lab contains the feature data for one data observation, i.e. the feature data for one store's sales in a particular week. To optimize the input pipeline, we want to convert each DataFrame row into a TensorFlow Example object. By using Example objects in the input pipeline, we're able to efficiently feed the data into a machine learning model.

After converting a DataFrame row to a TensorFlow Example, the row's integer valued features will be represented by Int64List TensorFlow Feature objects. From the analysis of our dataset, we know that the features with integer values are 'Store', 'Dept', 'IsHoliday', and 'Size'.



Time to Code!

In this chapter you'll be completing the add_int_features function, which adds all the integer features in a dataset row to the feature dictionary.

The features that contain integer values are: 'Store', 'Dept', 'IsHoliday', and 'Size'.

Set int_vals equal to a list containing the feature names with integer values.

For each integer valued feature, we'll create an Int64List containing the feature's value from dataset_row.

Create a for loop that iterates through int_vals using a variable named feature_name.

Inside the for loop, set list_val equal to tf.train.Int64List initialized with the value keyword argument set to a singleton list containing dataset_row[feature_name].

We can now map the feature's name to a TensorFlow Feature object representing its integer value.

Inside the for loop, set feature_name as a key in feature_dict. The value it maps to will be tf.train.Feature initialized with the int64_list keyword argument set to list_val.

