## **String Features**

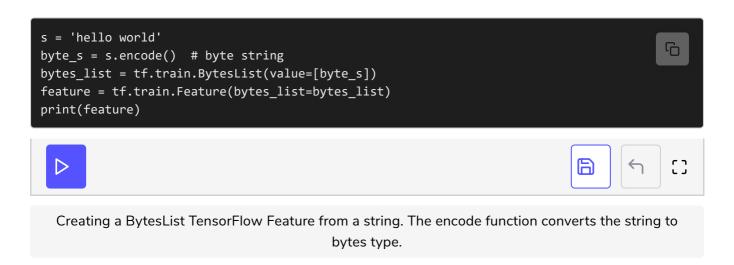
Learn about the string features used in the dataset.

## **Chapter Goals:**

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

## A. Adding string features

The third type of TensorFlow Feature object that can be used in an Example object is a <code>BytesList</code> TensorFlow Feature. This can represent either byte values (e.g. image data) or string values. For string values, we need to convert them to the <code>bytes</code> type prior to initializing a <code>BytesList</code> Feature object.



From the analysis of our dataset, we know that the only feature containing string values is 'Type'.

## Time to Code!

In this chapter you'll be completing the <a href="mailto:creates">create\_example</a> function, which creates an Example object from a row of the dataset.

We've already initialized a feature dictionary and added the integer and float features, using the functions from the previous two chapters. The only feature that contains string values is 'Type', so we need to convert the 'Type' value in dataset\_row to bytes, and then create a BytesList.

Set byte\_type equal to dataset\_row['Type'], converted to bytes.

Set list\_val equal to tf.train.BytesList initialized with the value keyword argument set to a singleton list containing byte\_type.

We can now complete the **feature\_dict** by mapping 'Type' to its corresponding TensorFlow Feature object.

Put 'Type' as a key in feature\_dict, and map it to a tf.train.Feature object initialized with the bytes\_list keyword argument set to list\_val.

Using the completed feature\_dict dictionary, we'll create and return a TensorFlow Example object containing the values in dataset\_row.

Set features\_obj equal to tf.train.Features initialized with the feature keyword argument set to feature\_dict.

Return a tf.train.Example object initialized with the features keyword argument set to features\_obj.

