Float Features

Learn about the float features used in the dataset.

Chapter Goals:

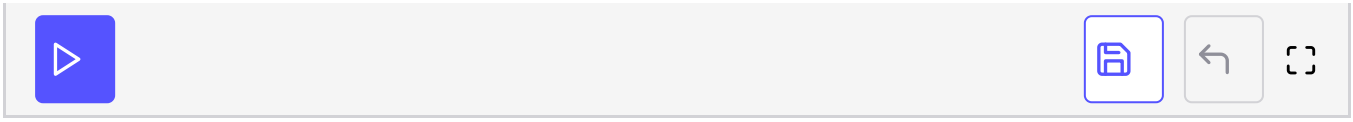
- Add the float features of a DataFrame's row to a feature dictionary

A. Adding float features

Similar to the integer valued features in the DataFrame row, we also convert the row's float valued features when creating the Example object. Each of the float features will be represented by `FloatList` TensorFlow Feature objects after the row is converted to an Example.

From the analysis of our dataset, we know that the float valued features are `'Temperature'`, `'FuelPrice'`, `'CPI'`, `'Unemployment'`, and `'Weekly_Sales'`.

```
print(feature_dict['Temperature'])
print(feature_dict['Fuel_Price'])
print(feature_dict['CPI'])
print(feature_dict['Unemployment'])
print(feature_dict['Weekly_Sales'])
```



The TensorFlow Feature objects for each of the float features in a single observation in the dataset (i.e. a row in the pandas DataFrame). The feature_dict dictionary maps string names to Feature objects.

The `'Weekly_Sales'` feature acts as the *label* used in training and evaluating the machine learning model. However, this means that the feature is not present outside of training/evaluation, i.e. when making real-life sales predictions with the model.

Therefore, we only include the `'Weekly_Sales'` feature in our Example objects for training and evaluation. In other words, only when the Example object should contain a label.

Time to Code!

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

The features that contain float values are: `'Temperature'`, `'Fuel_Price'`, `'CPI'`, `'Unemployment'`, and `'Weekly_Sales'`.

However, we only use the `'Weekly_Sales'` feature if `has_labels` is `True`. This is because the `'Weekly_Sales'` feature represents the *label* used in training/evaluating the machine learning model, which is not present when making predictions.

Set `float_vals` equal to a list containing the feature names with float values. Only include `'Weekly_Sales'` in the list if `has_labels` is `True`.

For each float valued feature, we'll create a `FloatList` containing the feature's value from `dataset_row`.

Create a `for` loop that iterates through `float_vals` using a variable named `feature_name`.
Inside the `for` loop, set `list_val` equal to `tf.train.FloatList` 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 float 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 `float_list` keyword argument set to `list_val`.

```
import tensorflow as tf

# Add the float Feature objects to the feature dictionary
def add_float_features(dataset_row, feature_dict, has_labels):
    # CODE HERE
    pass
```