

## Imputation of Missing Data

Another common need in feature engineering is handling missing data. We discussed the handling of missing data in DataFrames in “[Handling Missing Data](#)” on page 119, and saw that often the NaN value is used to mark missing values. For example, we might have a dataset that looks like this:

```
In[14]: from numpy import nan
        X = np.array([[ nan, 0,  3 ],
                      [ 3,  7,  9 ],
                      [ 3,  5,  2 ],
                      [ 4,  nan, 6 ],
                      [ 8,  8,  1 ]])
        y = np.array([14, 16, -1,  8, -5])
```

When applying a typical machine learning model to such data, we will need to first replace such missing data with some appropriate fill value. This is known as *imputation* of missing values, and strategies range from simple (e.g., replacing missing values with the mean of the column) to sophisticated (e.g., using matrix completion or a robust model to handle such data).

The sophisticated approaches tend to be very application-specific, and we won’t dive into them here. For a baseline imputation approach, using the mean, median, or most frequent value, Scikit-Learn provides the `Imputer` class:

```
In[15]: from sklearn.preprocessing import Imputer
        imp = Imputer(strategy='mean')
        X2 = imp.fit_transform(X)
        X2
```

```
Out[15]: array([[ 4.5,  0. ,  3. ],
                 [ 3. ,  7. ,  9. ],
                 [ 3. ,  5. ,  2. ],
                 [ 4. ,  5. ,  6. ],
                 [ 8. ,  8. ,  1. ]])
```

We see that in the resulting data, the two missing values have been replaced with the mean of the remaining values in the column. This imputed data can then be fed directly into, for example, a `LinearRegression` estimator:

```
In[16]: model = LinearRegression().fit(X2, y)
        model.predict(X2)
```

```
Out[16]:
array([ 13.14869292,  14.3784627 , -1.15539732,  10.96606197, -5.33782027])
```

## Feature Pipelines

With any of the preceding examples, it can quickly become tedious to do the transformations by hand, especially if you wish to string together multiple steps. For example, we might want a processing pipeline that looks something like this: