|  |  |
| --- | --- |
| **Label encoding** | **pandas.get\_dummies** |
| Cannot process string values directly. If your nominal features are strings, then you need to first map them into integers. | pandas.get\_dummies is kind of the opposite. By default, it only converts string columns into one-hot representation, unless columns are specified. |
| OHE does not add variables to your data frame | drawbacks. First, it modifies your dataframe |
| It creates dummy variables by transforming X, and all the dummies are stored in X | get\_dummies adds hundreds of dummy variables to the dataframe. |
| |  | | --- | | # Import label encoder | |  | from sklearn import preprocessing | |  | # label\_encoder object knows how to understand word labels. | |  | label\_encoder = preprocessing.LabelEncoder() | |  | # Encode labels in column 'Country'. | |  | data['Country']= label\_encoder.fit\_transform(data[‘Country']) | |  | print(data.head()) |   **Output**:  https://cdn.analyticsvidhya.com/wp-content/uploads/2020/02/table2-1.png  can see here, label encoding uses alphabetical ordering. Hence, India has been encoded with 0, the US with 2, and Japan with 1. | |  | | --- | | #importing the libraries | |  | import pandas as pd | |  | import numpy as np | |  |  | |  | #reading the dataset | |  | df=pd.read\_csv("Salary.csv") |   **Output**:  One Hot EncodingAs you   |  | | --- | | print df.info |   **Output**:  One Hot Encoding  As you can see here, the first column, Country, is the categorical feature as it is represented by the **object data type**and the rest of them are numerical features as they are represented by *int64*. |