**Data Pre-Processing**

Data preprocessing is a technique which is used to transform the raw data in a useful and

efficient format.

**Different forms:**

There are many different forms such as

* Structured Tables
* Images
* Audio files
* Videos etc..

In any Machine Learning process, Data Preprocessing is that step in which the data gets

transformed, or Encoded, to bring it to such a state that now the machine can easily parse it.

In other words, the features of the data can now be easily interpreted by the algorithm.

**Why need preprocessing?**

Since most of the real world data is raw, the data we collected from different resources such as the internet, self collection and organization needs to be preprocessed.

There may be problems in the dataset due to

* human error
* limitations of measuring devices
* flaws in the data collection process etc.

**Issues with the dataset**

Following are the issues with the dataset

* Missing values
* Inconsistent values
* Duplicate values etc.

We need to remove these kinds of things.

* **Missing values**

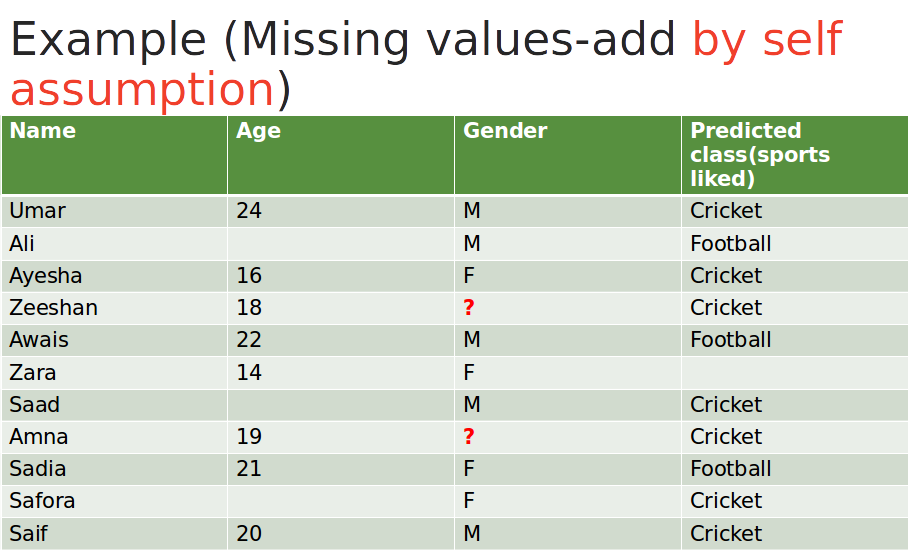
If our dataset has missing values we have to complete our data.

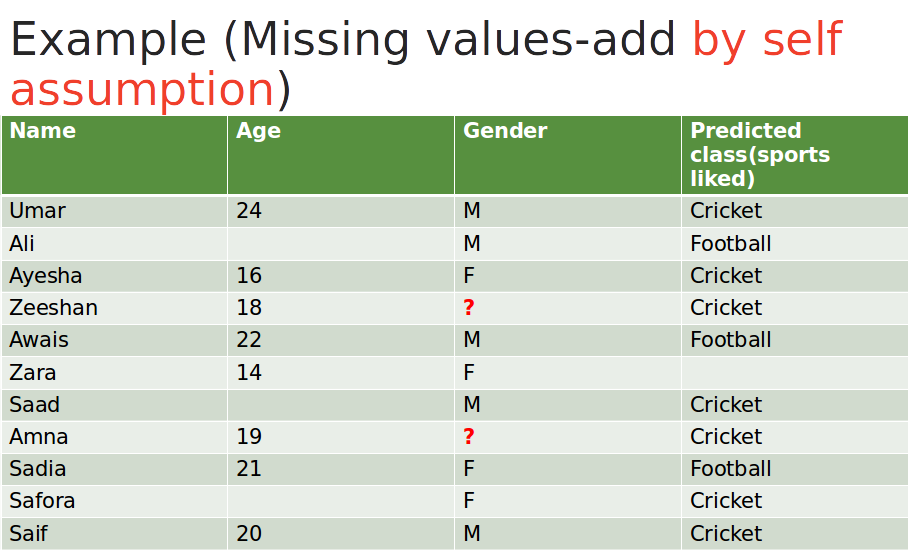
**How to handle missing values?**

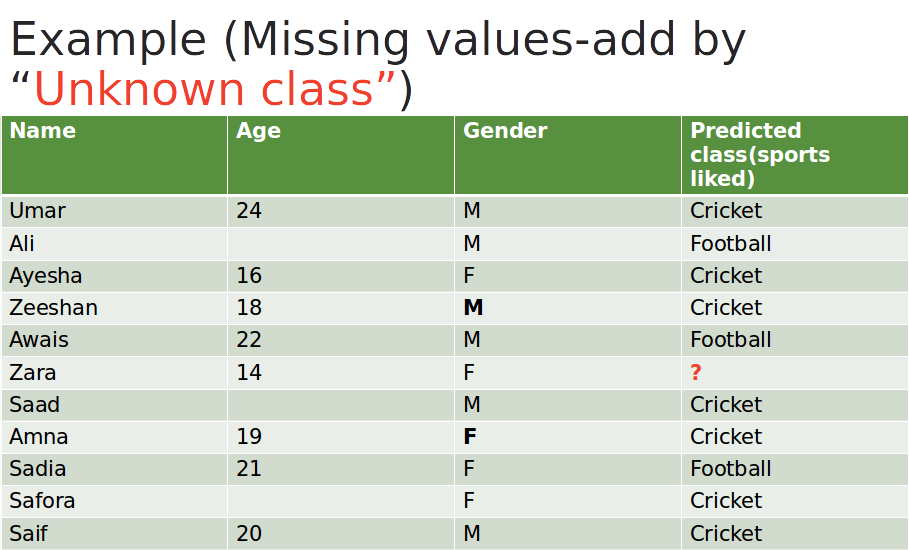
There are different approaches to handle the missing values

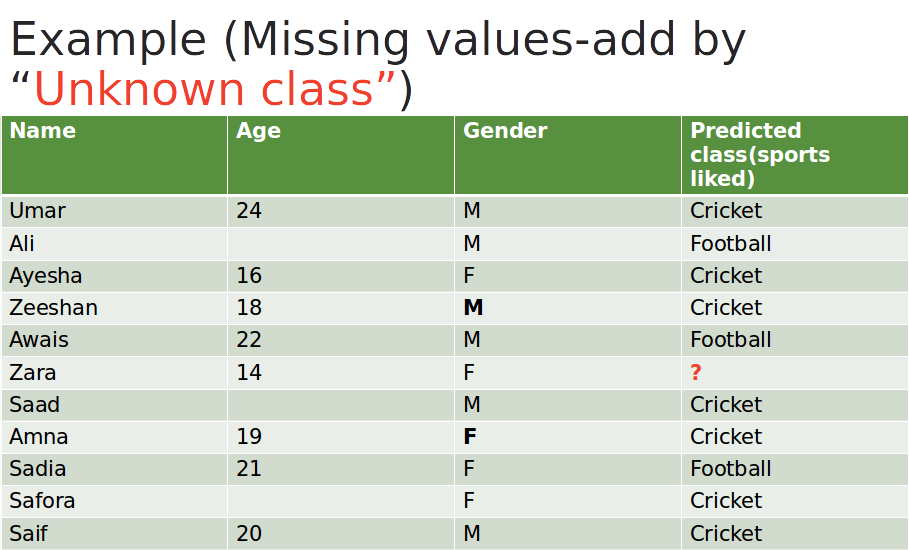
* *Case1:* remove the missing values
* Case 2: add the values by assumption. Not a good approach
* *Case 3:* add the other class “unknown”
* *Case 4:* Use the “mean” of all data and add it.

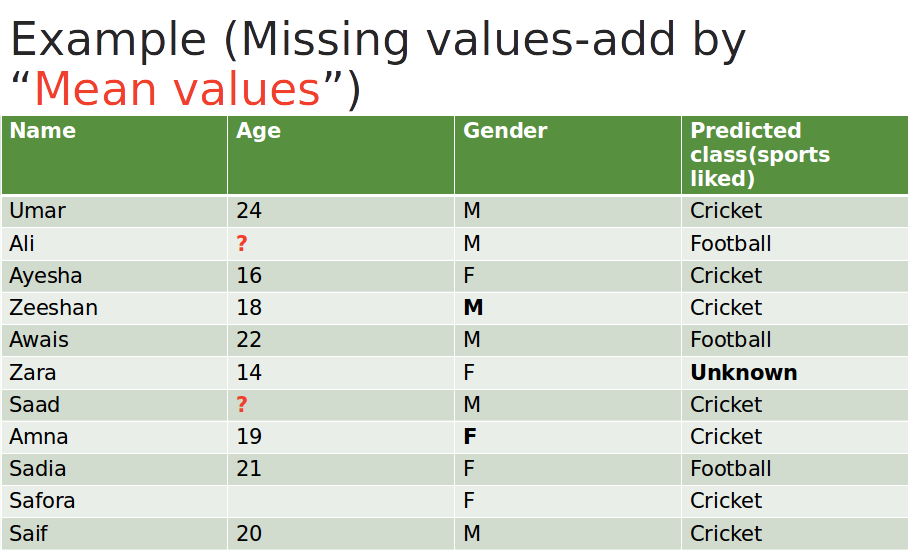


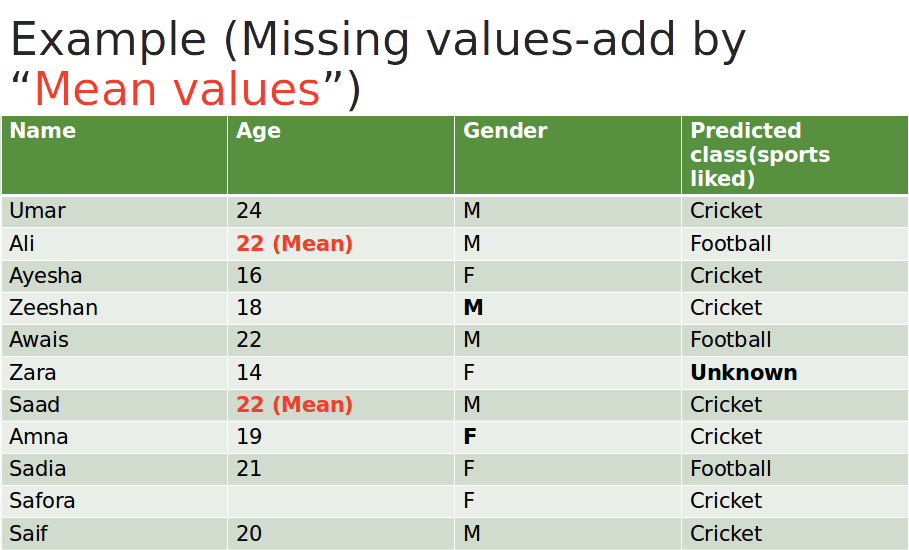












**Convert Categorical Data to Numerical Data**

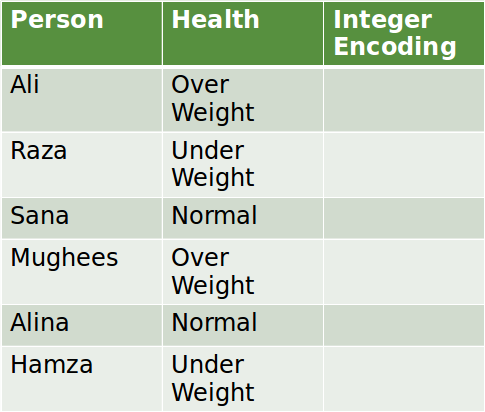
Data often contain label values rather than numeric values. Machine learning models algorithm required numerical data.

To overcome this we have 2 methods

* Integer Encoding
* One-Hot Encoding
* **Integer Encoding**

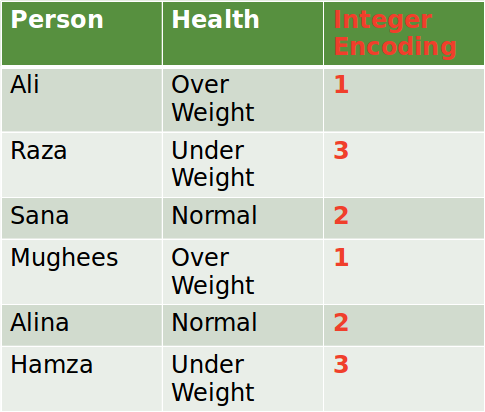
Each unique category value is assigned an integer value.

**Example 1**: person weight i.e. Overweight, Normal, and Underweight.



Each unique category value is assigned an integer value in the above example.

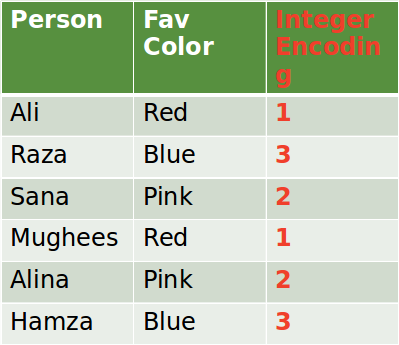
**Example:** person weight i.e. Over Weight is 1, Normal is 2, and Under Weight is 3.



**Example 2:** “Red” is 1, “Pink” is 2, and “Blue” is 3.

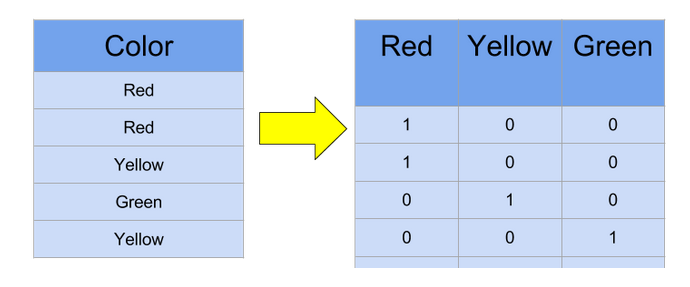
In this example no such ordinal relationship exists, the integer encoding is not enough. In fact, using this encoding and allowing the model to assume a natural ordering between categories may result in poor performance or unexpected results

(predictions halfway between categories).



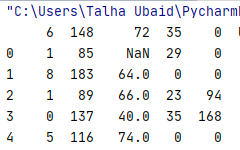
* **One-Hot Encoding**

One hot encoding creates new (binary) columns, indicating the presence of each possible value from the original data. Let's work through an example.



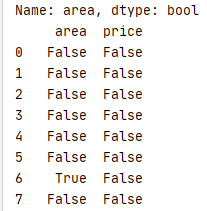
**To view the missing values in your dataset**

If our dataset has empty/ null columns. It will show “nan” in that cell when printed**.(using .head())**

****

To check the missing values : **(using .isnull())**

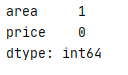
Here true values means null/empty values in there at that location.



**To count the null values in data**

To count the null values in data use the **.isnull().sum()**

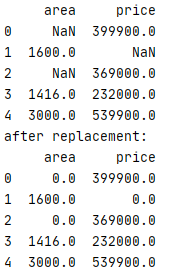
Here “area” and price are the columns and in the area there are only 1 missing values while in price we have no missing values.



**Replacing with values any values with “NaN”**

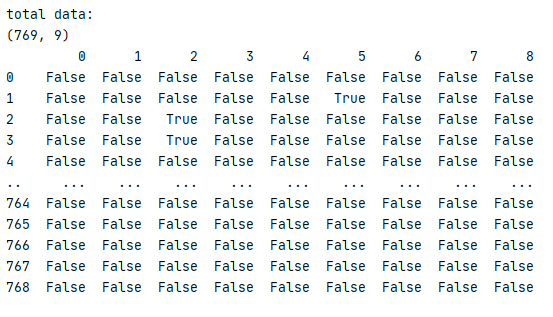
To replace 0 with nan use **.replace(nan,0)**

Here we can replace “nan” with any number or vice versa. We can replace it with any number.



**To remove row that has missing values**

To remove row that has missing values use .**dropna(inplace=true)**

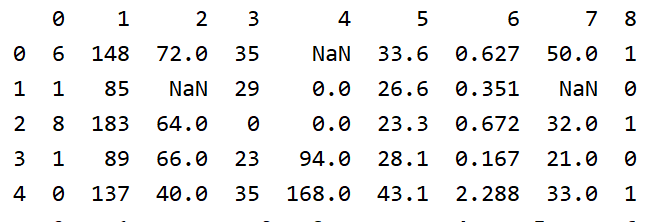
****

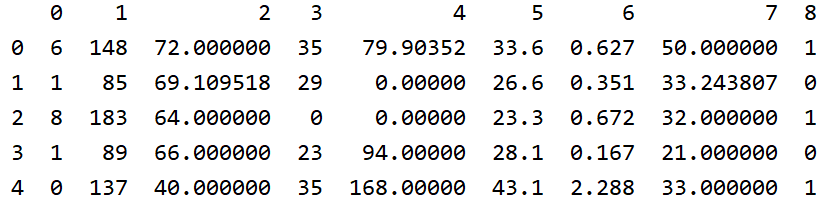
****

**To input missing values**

Multiples methods can be used such as mean, mode to input missing values

Use **.fillna(dataset.mean(),inplace=true)**

****

****