**Data pre-processing**

It’s the start of a new project and you’re excited to apply some machine learning models. You take a look at the data and quickly realize it’s an absolute mess. “According to IBM Data Analytics you can expect to spend up to 80% of your time cleaning data.”

**Missing values**

* Missing value occurs when no data is stored for a variable in an observation.
* Could be represented as NA or NaN or 0 etc

**Sources of missing values**

* User forgot to fill in a field.
* Data was lost while transferring manually from a legacy database.
* There was a programming error.
* Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

**How to Deal with them**

There are basically two kinds of missing values: Standard and Non-Standard missing values.

Standard missing values are those which pandas cannot detect such as empty blank in dataframe etc. whereas non-standard are which pandas can detect and replace with n/a or NaN.

Few ways to deal with them:

* Check with Data Collection source(person or group)
* Remove missing values:
  + Drop variable
  + Drop data entry
* Replace missing values:
  + Replace it with average value (for numbers)
  + Replace it with frequency (for non-numeric)
  + Replace it based on other functions or trends (like which tend to seem possible after looking at the whole data)
* Leave it as a missing data.

**Drop missing values**

* -dropna(): Drop missing values. It will drop entire row or column as whole
* axis = 0 -> drop entire row
* = 1 -> drop entire col
* df.dropna(susbset=["Price"], axis=0,inplace=True)
* df.replace(missing\_value , new\_value)

**Data Formatting**

Data is usually collected from different places by different people which may be stored in different formats. Data formatting means bringing data into a common standard of expression that allows users to make meaningful comparisons. As a part of dataset cleaning, data formatting ensures the data is consistent and easily understandable.

It is important for later analysis to explore the features data type and convert them to the correct data types. Otherwise, the developed models later on may behave strangely, and totally valid data may end up being treated like missing data.

* To identify datatypes: df.dtypes()
* To convert datatypes: df.astype()

**Data Normalization**

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. For machine learning, every dataset does not require normalization.

Several approaches for Data Normalization:

* Simple feature scaling: x(new) = x(old)/x(max)
* Min-max: x(new) = (x(old)-x(min))/(x(max)-x(min))
* Z-Score: x(new) = (x(old) - mean)/SD

**Binning**

Data binning, bucketing is a data pre-processing method used to minimize the effects of small observation errors. The original data values are divided into small intervals known as bins and then they are replaced by a general value calculated for that bin. This has a smoothing effect on the input data and may also reduce the chances of overfitting in case of small datasets

**Turning categorical variables into quantitative**

Most statistical models cannot take in objects or strings as input and for model training only take the numbers as inputs. There are two ways for this purpose. They are:

* One-hot encoding: One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.
* get\_dummies() : In Pandas, we can use get\_dummies method to convert categorical variables to dummy variables. In Python, transforming categorical variables to dummy variables is simple. The get\_dummies method automatically generates a list of numbers, each one corresponding to a particular category of the variable.