**Effective techniques of Data Cleaning for Machine learning**

The most important and major step of machine learning algorithms is Data Cleaning.If the data is not cleaned properly then the insights produced by will not be correct.

For a data to be used for machine learning algorithms it is mandatory that the data should be only in numerical format and in structured format(tables).

The data we get to do modelling for ML algorithms is got through internet or many other [ways. It](http://ways.it/) is not sure that the data will always be in the format that we require.

So it is mandatory to clean the data for the above reasons.

The problems which occur if we not clean the data are listed below:

1. Missing data or null values.
2. Unrequired data.
3. Duplicate values.
4. Incorrect format of data.
5. Outliers.
6. Unstructured data.
7. Textual data problems.
   * Spelling mistakes.
   * Lowercase or Uppercase issues.
   * Unwanted whitespaces.
   * Special characters.
   * Different languages in data.

**1.Missing data or null values**

If the data has missing values or null values,the best way to handle this is to either drop or impute.

**df=df.dropna( )#Delete the rows which has NaN values**

But often we can't drop the data as it may lead to information loss. In such cases we will impute the data.

For categorical data, we replace the NaN or missing values by mode of the particular column.

For numerical data, we replace the NaN or missing values by mean or median of the particular column.

**df.isnull( ).sum( ) gives the column names and count of the column names.**

**x=df['Age'].mean( )#Finding the mean of age column.**

**df['Age']=df['Age'].fillna(x)£Filling the NaN values with the mean of age.**

**2.Duplicate values**

If the dataset has same rows then it will be difficult to predict correct results through ML algorithms.

**df=df.drop\_duplicates( )#Drops the repeating rows in a dataframe**

**3.Incorrect format of data**

In the dataset some of the integer columns will have a datatype as [objects. In](http://objects.in/) such cases we need to change the datatype to integer .

**df.dtypes #Gives the column names and its Datatype**

**df['Age']=df['Age'].astype('int64')**

If the dataset has string datatype then the encoding should be done based on the type of data.

**df["Purchased"]=df["Purchased"].map({'Yes':1,'No':0})**

In the above example the purchased coulumn has only two values Yes/No.Yes is converted to 1 and No is converted to 0.

Now the data is completely converted to numerical format.

**4.Unrequired data**

If we think the data or any columns or any rows are unnecessary then we can drop it.

**df=df.drop(['B'],axis=1)#Drops the column named 'B'.**

**5.Outliers detection**

Outliers are extremely high or low values.

It is necessary to detect outliers.

**Interquantile Range(IQR)=Q3-Q1**

If any value is greater than Q3+(1.5\*IQR) then the dataset has outliers.

If any value is less than Q1-(1.5\*IQR) then the dataset has outliers.

If outliers are detected in a dataset then clipping must be applied.

In clipping all the values greater than upperthreshold is brought to upper\_threshold values and all the values less than lowerthreshold is brought to lower\_threshold.

**upper\_threshold = df2['Brain'].quantile(0.75) + (1.5 \* iqr)**

**lower\_threshold = df2['Brain'].quantile(0.25) - (1.5 \* iqr)**

**upper\_threshold,lower\_threshold**

**Output:**

(119.60625, -70.80375000000001)

**df2["Brain"]=df2["Brain"].clip(-70,119) #Clipping the data.**

Most of the Data Scientists spend 70-80% of their time in Data Cleaning.