**DATA PREPROCESSING-DATA CLEANING**

**#DROP MISSING VALUES**

import pandas as pd

train=pd.read\_csv("train1.csv")

***Note: create any dataset with missing entry (5 attributes with 50 populations)***

**#This will output the first x rows of our data. Viewing this will give one a quick view on the presence of NaN/-1/0/blank/? among others.**

train.head(10)

**#This will tell us the total number of non null observations present including the total number of entries. Once number of entries isn’t equal to number of non null observations, we can begin to suspect missing values**

train.info()

**#This will tell us the total number of NaN in or data.**

train.isnull().sum()

**#This is the fastest and easiest step to handle missing values. However, it is not generally advised. This method reduces the quality of our model as it reduces sample size because it works by deleting all other observations where any of the variable is missing. The process can be done by:**

df=train.dropna()

df.info()

**#It will be observed that of 891 entries will be reduced to 183 just by dropping NaN values.!!!**

**Dropping is only advised to be used if missing values are few (say 0.01–0.5% of our data).**

**#FILLING MISSING VALUES**

**#If the missing value isn’t identified as NaN , then we have to first convert or replace such non NaN entry with a NaN.**

train['Age'].replace(0, np.NaN)

**#Age is a column name for our train data**

mean\_value=train['Age'].mean()

**#this will replace all NaN values with the mean of the non null values**

train['Age']=train['Age'].fillna(mean\_value)

**#check whether age attribute have null value or not**

**train.isnull().sum()**

**#For Median**

median\_value=train['Age'].median()

train['Age']=train['Age'].fillna(median\_value)