**How to deal with Missing Values:** **Missing Value Imputation**

There are 4 ways for Missing Value Imputation:

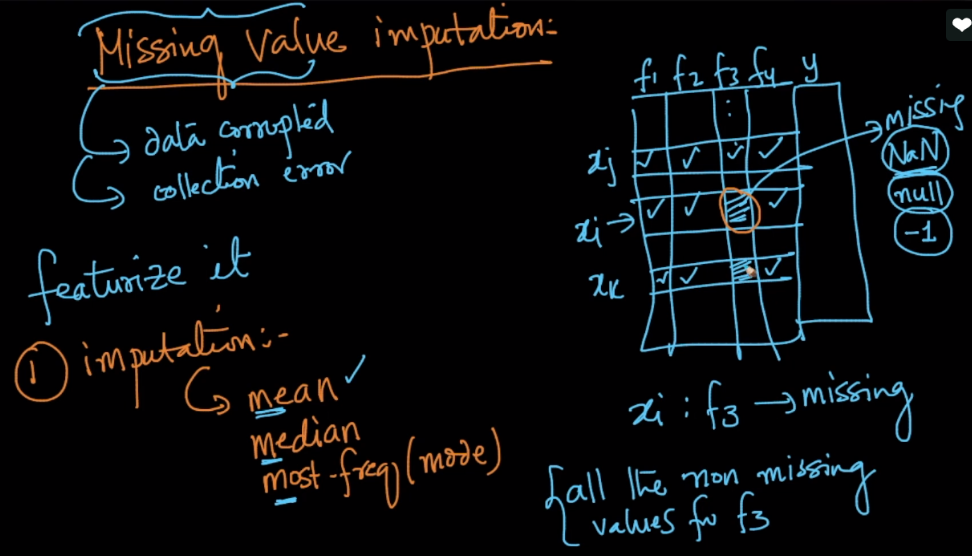
**1st Approach:** **Simple Imputation**

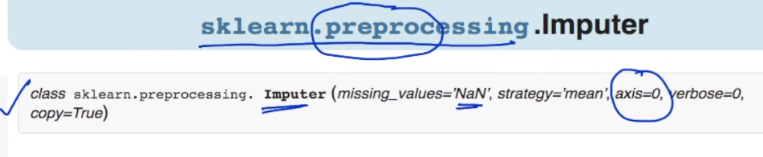
In this technique we simply replace the missing value with mean, median or mode value of that column or feature.

If feature contains **numeric value**, then we can replace using **Mean, Median.**

If feature contains **text value**, then we can replace using **Most-frequent term(Mode)**

Example, let’s say there are missing value in feature f3, so what we do is take all the non-missing value of f3, and find it’s Mean, Median or Mode and replace missing value with that.



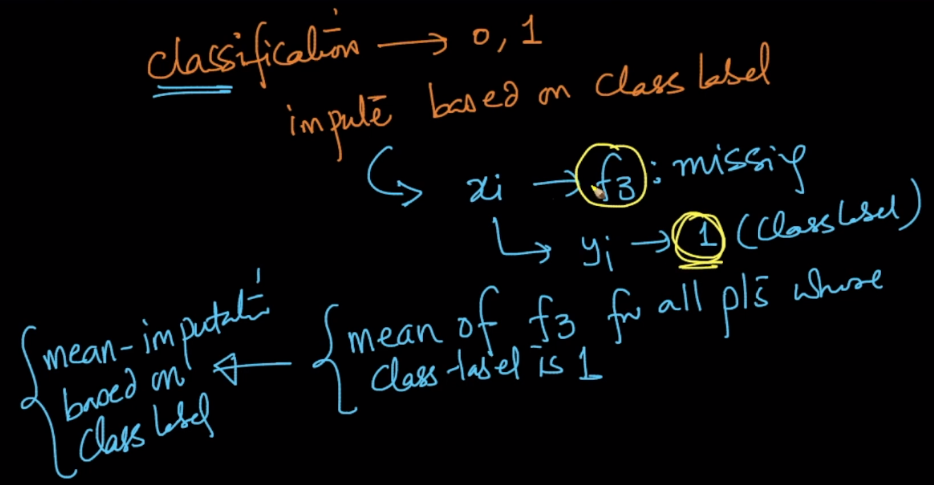


**2nd Approach: Imputation based on class label.**

In 1st approach we were taking mean or median or mode of all the non-missing values without considering the output or class label that non-missing value is corresponding.

So in this approach what we do is, let’s say a value x is missing in feature f3, whose output or class label is 1.

So now we take only those non-missing values from feature f3, whose class label is one, and calculate mean for them and replace in missing value.

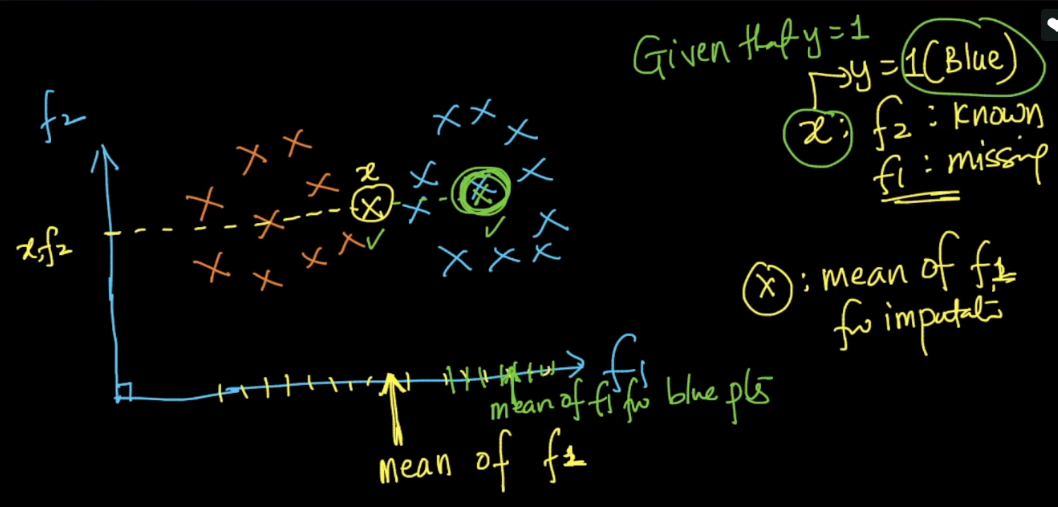


**Why we are doing this?**

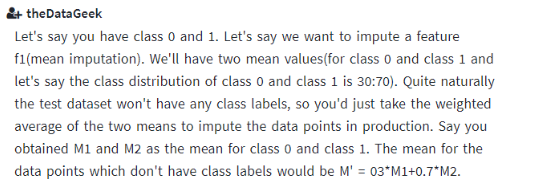
Let’s suppose we have dataset with 2-D or 2 features whose output is classified as 0 or 1, we know f2 for a data point, but f1 is missing.

Now we can directly calculate mean of non-missing values of f1, and replace in missing value. But this will give the point in between of that two clusters.

Now if we do class based imputation, let’s say missing value data point is referring to class label 1, then we’ll only take non-missing value in f1 with class label 1, so it will the value within the cluster whose label is 1.



**How we can do class label imputation for new data point, which is given as an input to predict it’s output. So now we don’t know class label of this data point.**



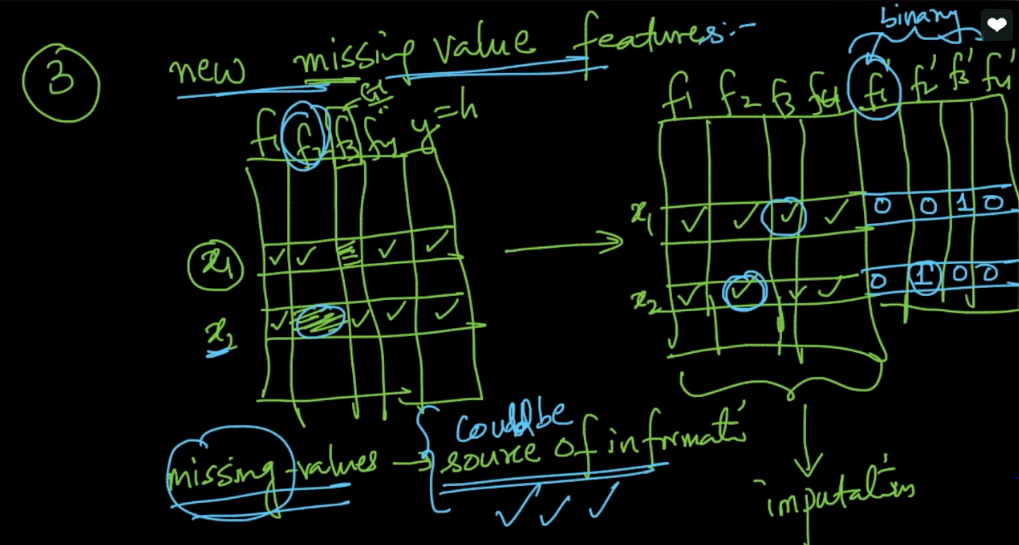
**3rd Approach: New Missing Value Features:**

In this approach we simply creates new features which corresponds to all the old features, whose value will be 1, if that features’ value is missing for that row, otherwise it will be 0.

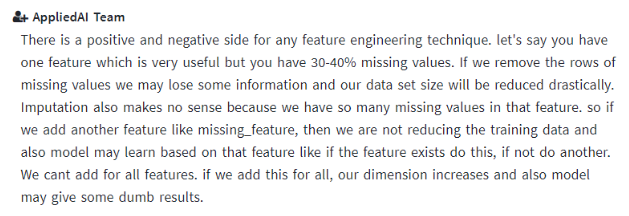
Suppose given a dataset whose f3 value is missing for x1 row, and f2 value is missing for x2 row.

Now since we’ve 4 features(f1, f2, f3, f4), so we create 4 new features(f1’, f2’, f3’, f4’), which contains only 1 and 0.

Since f3 is missing in row x1, so that cell value will be 1 and rest cell values in x1 will be 0. And same for f2 in row x2



**But this approach increase dimensions drastically, so where should we use it?**



**4th Approach: Model Based Imputation**

Let’s say we’ve dataset in which f3 feature has missing value. Now In model based Imputation what we do is.

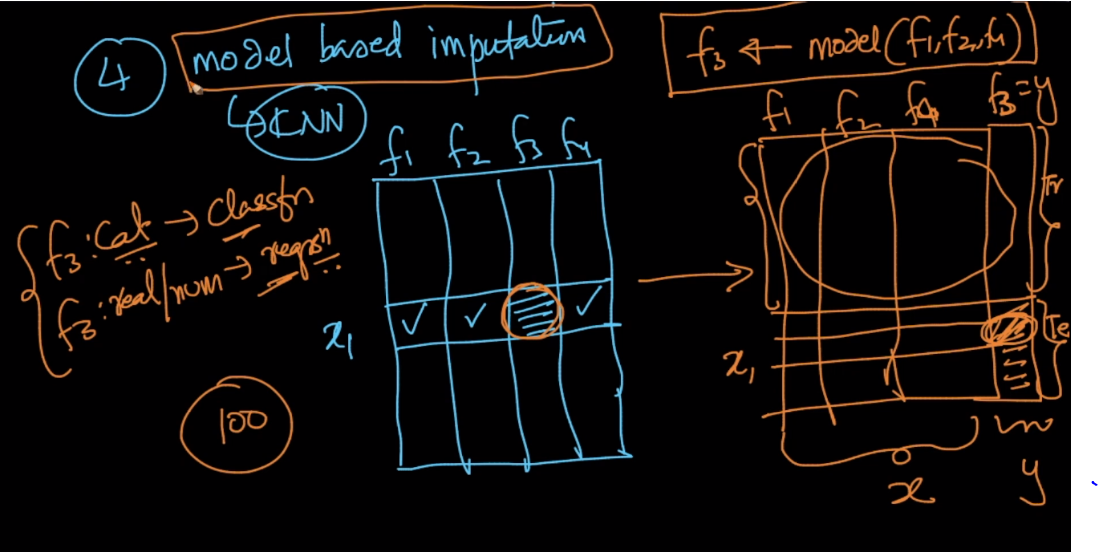
Suppose initially we’ve features (f1, f2, f3, f4).

Now we’ll train model using (f1, f2, f4) as input features and f3 as output.

We’ll divide this new dataset into train and test data, as all the rows in which f3 values are non-missing will be put into train data, hence it would be used to train model.

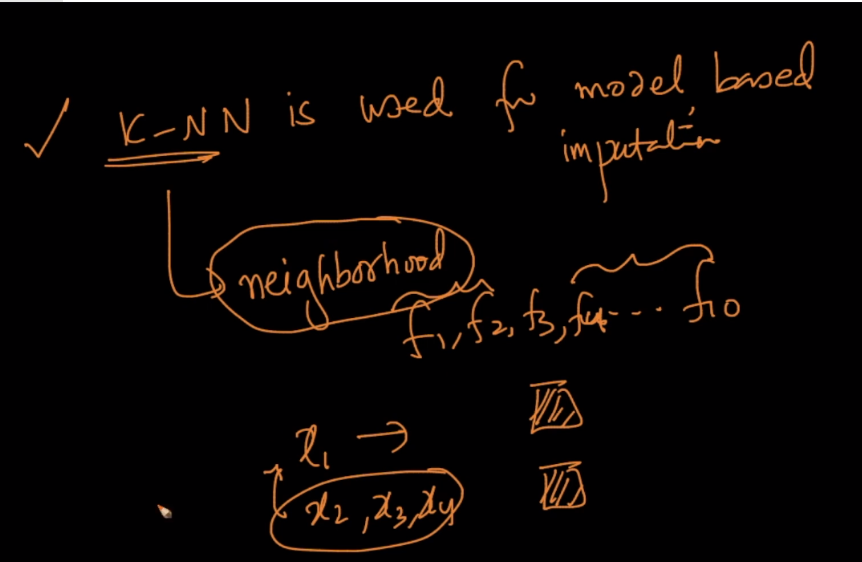
And all the rows whose f3 values are missing will be put into test data, because we want to predict value for this non-missing values.

And we train model using new train dataset, and when test data is applied on it, it will give the output for each missing value of f3(the output can categorical for classification and numeric for Regression).



**K-NN** is mostly used for model based imputation.

Why?, because since in K-NN we use concept of neighborhood, and the missing value also would similar to it’s neighbors.



<https://machinelearningmastery.com/handle-missing-data-python/>