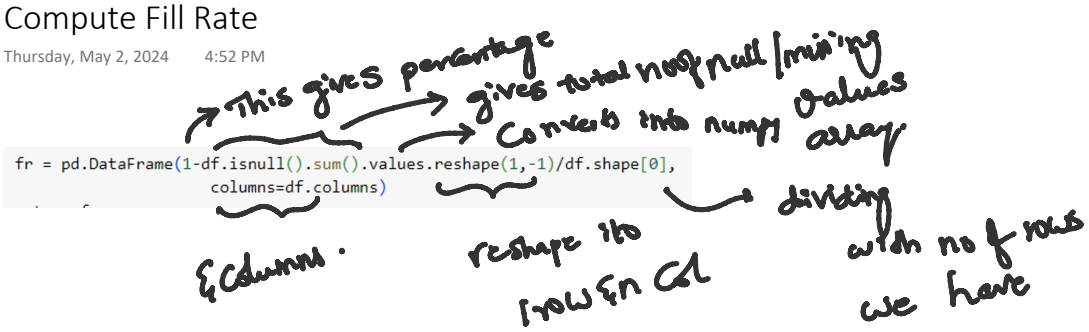
Compute Fill Rate

Thursday, May 2, 2024 4:52 PM




```
fr = pd.DataFrame(1-df.isnull().sum().values.reshape(1,-1)/df.shape[0],
                  columns=df.columns)
```

- Calculate the fill rate for each feature (column) in your dataset.
- For each feature, count the number of non-missing values and divide it by the total number of observations. Multiply by 100 to get the fill rate percentage.
- Based on the computed fill rates, decide how to handle missing values. If a feature has a high fill rate (close to 100%), you might choose to drop rows or columns with missing values. For features with a low fill rate, you might impute missing values using methods like mean, median, mode imputation, or more sophisticated techniques like regression imputation or K-nearest neighbors imputation.
- For example if we got 1.0 for a column after calculating compute fill rate that means there are no missing values and all the data is populated in the data frame
- Suppose we have got 26.12 that mean around 26 percent of missing values are present and only 74 percent of data we have.

```
def compute_fill_rate( df ) :
    """
    Computing the rate of non-NaNs for each column
    Params :
        df : Pandas dataframe, input data
    Return :
        Pandas dataframe
    """
    fr = pd.DataFrame(1-df.isnull().sum().values.reshape(1,-1)/df.shape[0],
                      columns=df.columns)
    return fr
```

```
compute_fill_rate( df_norm )
```

	Sr. No.	Location	Sub-Area	Property Type	Property Area in Sq. Ft.	Price in Millions	Company Name	TownShip Name/ Society Name	Total TownShip Area in Acres	ClubHouse	School / University in Township	Hospital in TownShip	Mall in TownShip	Park / Jogging track	Swimming Pool
0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.260204	1.0	1.0	1.0	1.0	1.0	1.0

So input ==>Data frame
output ==> Compute Fill rate values for each column (0,1) 1 indicates no missing values.