**Credit Risk Modelling – A tutorial**

**Gross Credit loss = Contractual charge off loss + bankruptcy loss**

Charge off loss is a case when a consumer does not pay the money let’s say after n days of the due date that has already passed

Bankruptcy loss occurs when a consumer declares himself as bankrupt and cannot pay any money back to the lender on day 0 itself. In that case the lender is in no position to collect the money from the consumer

And then the debt asset sale and recoveries are done for such delinquent accounts, if we look into this picture the net credit loss is nothing but:

**NET CREDIT LOSS = GROSS CREDIT LOSS – RECOVERIES – ASSET SALE -1**

**PD** – the probability of default : the likelihood that of a default in case when a account in contractually charged off or returned in case of bankruptcy. PD is always associated with a time horizon. For ex: if the PD for a certain account is 10% for a 12 month period, then it means that the same account can go default i.e without making any payments in the 12 month period.

**EAD** – Exposure at default : Estimate of exposure a bank has when the borrower defaults at the time of charge off or Bankruptcy

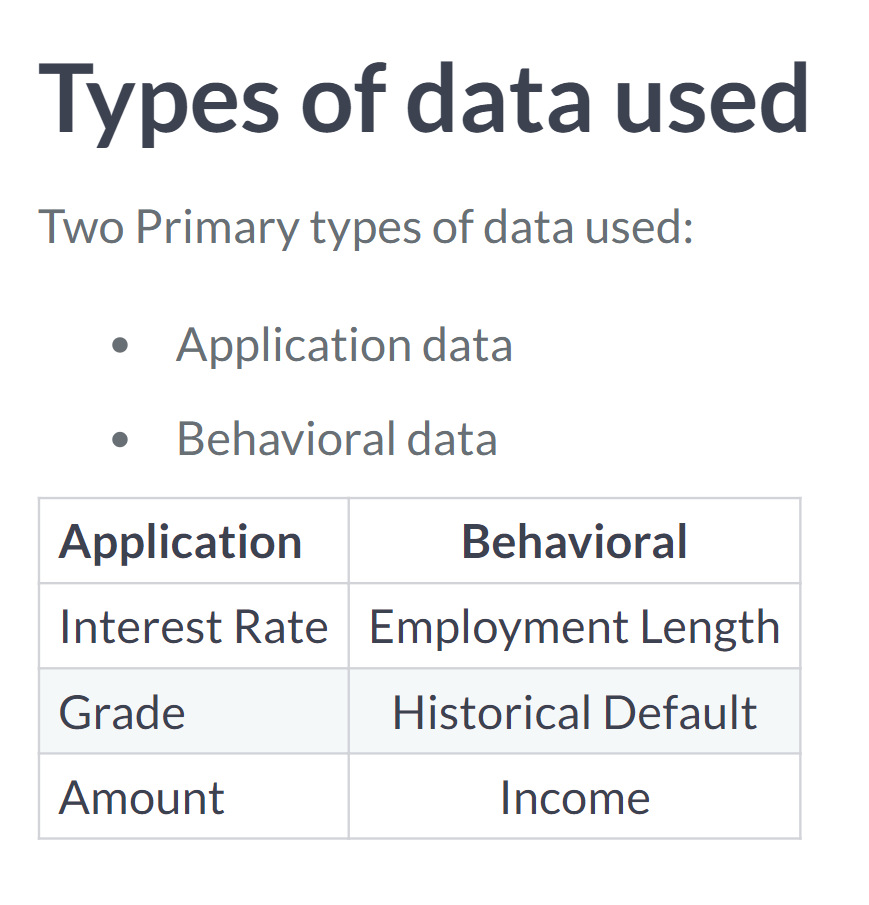
**LGD** – loss given default- It is directly related to the asset sale and recoveries. It is the % of exposure at the time of default that is eventually lost by the bank.

**GCL – Gross Credit Loss : PD\*EAD**

**NCL or expexted net loss – Net credit loss : PD\*EAD\*LGD or the equation 1 above**

Lets understand the modelling behind these components

PD – for PD we need two types of data as mentioned below:



**Data science for Credit Risk**

1.Exploring data with cross tabs and Pivots

2.Removing outliers

3.Handling missing values

How to handle missing data

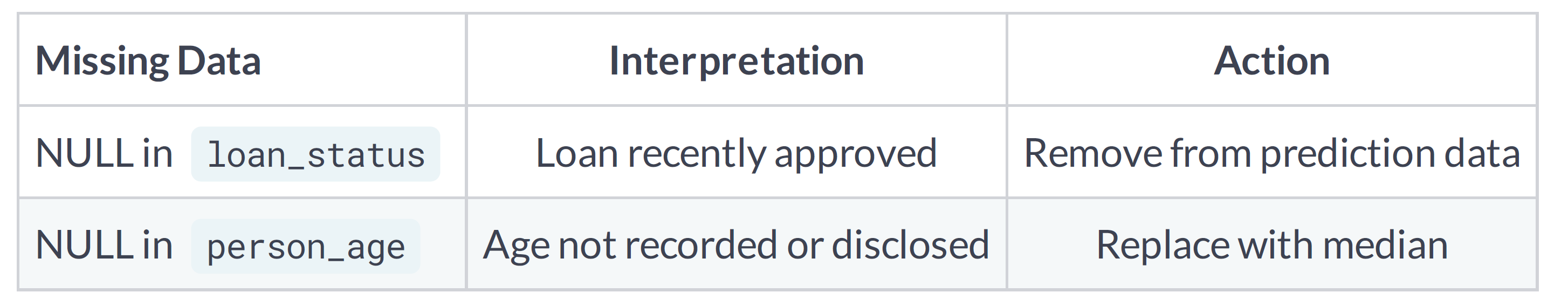
**Generally three ways to handle missing data**

Replace values where the data is missing

Remove the rows containing missing data

Leave the rows with missing data unchanged

Understanding the data determines the course of action:



**Finding missing data**

Null values are easily found by using the isnull() function

Null records can easily be counted with the sum() function

.any() method checks all columns

null\_columns = cr\_loan.columns[cr\_loan.isnull().any()]

cr\_loan[null\_columns].isnull().sum()

**# Replace the null values with the median value for all employment lengths**

# Print an array of columns with null values

print(cr\_loan.columns[cr\_loan.isnull().any()])

# Print the top five rows with nulls for employment length

print(cr\_loan[cr\_loan['person\_emp\_length'].isnull()].head())

cr\_loan['person\_emp\_length'].fillna((cr\_loan['person\_emp\_length'].median()), inplace=True)

# Create a histogram of employment length

n, bins, patches = plt.hist(cr\_loan['person\_emp\_length'], bins='auto', color='blue')

plt.xlabel("Person Employment Length")

plt.show()

**You can use several different functions like mean() and median() to replace missing data. The goal here is to keep as much of our data as we can! It's also important to check the distribution of that feature to see if it changed**

* To get the count of nulls, call the .sum() method.
* The attribute .index is what selects the index of each row.
* The .drop() method needs .index values of the rows with missing data.

**# Print the number of nulls**

print(cr\_loan['loan\_int\_rate'].isnull().sum())

# Store the array on indices

indices = cr\_loan[cr\_loan['loan\_int\_rate'].isnull()].index

# Save the new data without missing data

cr\_loan\_new = cr\_loan.drop(indices)

**##Now that the missing data and outliers have been processed, the data is ready for modeling! More often than not, financial data is fairly tidy, but it's always good to practice preparing data for analytical work.**

**To handle missing value data for a string col like person\_home\_ownership we do the following**

# Count the number of records for each unique value

cr\_loan['person\_home\_ownership'].value\_counts()