



# Data Cleaning and Preparing data for Analysis

---

Venkat Reddy Konasani

# Contents

- Model Building Life cycle
- Raw Data - issues
- Data Exploration
- Data Validation
- Data sanitizations techniques

# Model Building Life Cycle

## Background and Objective

Business Objective

Set Goals

Project Plan

Budget & Resources

## Data Exploration

Collect data

Explore data

Basic Summary

Identify outliers and missing values

## Preparing data for analysis

Validate data

Outlier treatment

Missing value treatment

Clean the data

Prepare data for Analysis

## Building the model

Select the right model

Variable selection

Model building and finetuning

Model iterations

## Validating the model

Intime validation

Out of time validation

Model finetuning

## Deployment

Deploy model

Maintenance of model

Model monitoring

# Data Exploration and Validation

- Junk in Junk out
- The final model is as good as the input data.
- You can never build a great model on erroneous input data

# Model Building Life Cycle

## Background and Objective

Business Objective

Set Goals

Project Plan

Budget & Resources

## Data Exploration

Collect data

Explore data

Basic Summary

Identify outliers and missing values

## Preparing data for analysis

Validate data

Outlier treatment

Missing value treatment

Clean the data

Prepare data for Analysis

## Building the model

Select the right model

Variable selection

Model building and finetuning

Model iterations

## Validating the model

In time validation

Out of time validation

Model finetuning

## Deployment

Deploy model

Maintenance of model

Model monitoring



# Raw Data - issues

---

# The raw data is dirty

- Wrong formats- expenses is read as date
- Might have missing values - Income missing for some records
- Might have outliers - Number of loans is 25000
- Erroneous values - Age is less than 0
- Default values - Account tenure is 999999
- Inconsistent - Age is 25, year of birth is 1970

# Preparing data for analysis

- We can't directly start the analysis and model building with raw data.
- Before getting on to core analysis and strategy building it is very important to
  - Explore the data
  - Validate the data
  - And finally clean the data and prepare it for analysis





# Case Study- Data Exploration

---

# Give me some credit data

- We will try to understand the data exploration, validation and data cleaning using a case study on loans data
- Give me some credit data. It is loans data. Historical data are provided on 150,000 borrowers.
- The final objective is to build a model that borrowers can use to help make the best financial decisions.
- We generally get the data and data dictionary from the data team.

# Data Dictionary

No	Variable Name	Short Description	Description	Varibale Type
1	SeriousDlqin2yrs	Target Variable (loan defaulter)	Person experienced 90 days past due delinquency or worse	Y/N
2	RevolvingUtilizationOfUnsecuredLines	Credit Utilization	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits	percentage
3	age	Age	Age of borrower in years	integer
4	NumberOfTime30-59DaysPastDueNotWorse	One month late frequency	Number of times borrower has been 30-59 days past due but no worse in the last 2 years.	integer
5	DebtRatio	Debt to income ratio	Monthly debt payments, alimony, living costs divided by monthly gross income	percentage
6	MonthlyIncome	Income	Monthly income	real
7	NumberOfOpenCreditLinesAndLoans	Number of loans	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)	Integer
8	NumberOfTimes90DaysLate	Three months late frequency	Number of times borrower has been 90 days or more past due.	integer
9	NumberRealEstateLoansOrLines	House loans	Number of mortgage and real estate loans including home equity lines of credit	integer
10	NumberOfTime60-89DaysPastDueNotWorse	Two months late frequency	Number of times borrower has been 60-89 days past due but no worse in the last 2 years.	integer
11	NumberOfDependents	Dependents	Number of dependents in family excluding themselves (spouse, children etc.)	integer

# Steps in Data Exploration and Cleaning

- Step-1: Basic details of the data
- Step-2: Categorical variables exploration
- Step-3: Continuous variables exploration
- Step-4: Missing Values and Outlier Treatment



# Step-1: Basic details of the data

---

# Check the Metadata

- Metadata is data about data
- What are total number of observations and variables
- Check each field name, field type, Length of field
- Are there some variables which are unexpected say q9 r10?
- Are the data types and length across variables correct
- For known variables is the data type as expected (For example if age is in date format something is suspicious)

# Print first few records

- Do we have any unique identifier? Is the unique identifier getting repeated in different records?
- Do the text variables have meaningful data?
- Are there some coded values in the data
- Do all the variables appear to have data? Are there any missing values

# Lab: Basic contents of the data

- Import “Give me some Credit\cs-training.csv”
- What are number of rows and columns
- Are there any suspicious variables?
- Are all the variable names correct?
- Display the variable formats
- Print the first 10 observations
- Do we have any unique identifier?
- Do the text and numeric variables have meaningful data?
- Are there some coded values in the data?
- Do all the variables appear to have data



# Code: Basic contents of the data

```
import pandas as pd
loans=pd.read_csv("D:\\Google Drive\\Training\\Datasets\\Give me
some Credit\\cs-training.csv")
loans
```

```
#What are number of rows and columns
loans.shape
```

```
#Are there any suspicious variables?
loans.columns.values
```

```
#Display the variable formats
loans.dtypes
```

# Code: Basic contents of the data

```
#Print the first 10 observations  
loans.head(10)
```

```
#Do we have any unique identifier?  
loans.columns.values
```



# Step-2: Categorical and Discrete variables exploration

---

# The Frequency Table and summary

- Calculate frequency counts cross-tabulation frequencies for Especially for categorical, discrete & class fields
- Frequencies
  - help us understanding the variable by looking at the values it's taking and data count at each value.
  - They also helps us in analyzing the relationships between variables by looking at the cross tab frequencies or by looking at association

# Check Points

1. Are values as expected?
2. Variable understanding : Distinct values of a particular variable, missing percentages
3. Are there any extreme values or outliers?
4. Any possibility of creating a new variable having small number of distinct category by clubbing certain categories with others.

# Lab: Frequencies

- What are the categorical and discrete variables? What are the continues variables.
- Find the frequencies of all class variables in the data
- Are there any variables with missing values?
- Are there any default values?
- Can you identify the variables with outliers?
- Are there any variables with other issues?

# Code: Frequencies

```
#What are the categorical and discrete variables? What are the continues variables.
```

```
loans.dtypes  
loans.head()
```

```
#Find the frequencies of all class variables in the data  
loans.columns.values
```

```
#Find the frequencies of all class variables in the data  
loans['SeriousDlqin2yrs'].value_counts()
```

```
loans['age'].value_counts(sort=False)
```

```
loans['NumberOfTime30-59DaysPastDueNotWorse'].value_counts(sort=False)
```

```
loans['NumberOfOpenCreditLinesAndLoans'].value_counts(sort=False)
```

# Code: Frequencies

```
loans[ 'NumberOfTimes90DaysLate' ].value_counts(sort=False)
```

```
loans[ 'NumberRealEstateLoansOrLines' ].value_counts(sort=False)
```

```
loans[ 'NumberOfTime60-89DaysPastDueNotWorse' ].value_counts(sort=False)
```

```
loans[ 'NumberOfDependents' ].value_counts(sort=False)
```

```
#Are there any variables with missing values?
```

```
loans.isnull().sum()
```





# Step-3: Continuous variables exploration

---

# Summary of Continuous variables

- Min, Max, Median, Mean, sd, Var
- Quartiles
- Box plots and identification of outliers
- Percentiles- P1, p5,p10,q1(p25),q3(p75), p90,p99

# Check Points

- Are variable distribution as expected.
- What is the central tendency of the variable? Mean, Median and Mode across each variable
- Is the concentration of variables as expected ? What are quartiles?
- Indicates variables which are unary I.e stddev=0 ; the variables which are useless for the current objective.
- Are there any outliers / extreme values for the variable?
- Are outlier values as expected or they have abnormally high values - for ex for Age if max and p99 values are 10000. Then should investigate if it's the default value or there is some error in data
- What is the % of missing value associated with the variable?

# LAB: Continuous variables summary

- List down the continuous variables
- Find summary statistics for each variable. Min, Max, Median, Mean, sd, Var
- Find Quartiles for each of the variables
- Create Box plots and identify outliers
- Find the percentage of missing values
- Find Percentiles and find percentage of outliers, if any P1, p5, p10, q1(p25), q3(p75), p90, p99

# Code: Continuous variables summary

## 'RevolvingUtilizationOfUnsecuredLines

```
#Find Percentiles and find percentage of outliers, if any P1,  
p5,p10,q1(p25),q3(p75), p90,p99
```

```
util_percentiles=loans['RevolvingUtilizationOfUnsecuredLines'].quantil  
e([0.05, 0.1, 0.25, 0.5, 0.75, 0.80,  
0.9,0.91,0.95,0.96,0.97,0.975,0.98,0.99,1])round(util_percentiles,2)
```

# Code: Continuous variables summary

## MonthlyIncome

```
#Find the percentage of missing  
valuesloans['MonthlyIncome'].isnull().sum()loans['MonthlyI  
ncome'].isnull().sum()/len(loans)
```



# Data Cleaning

---

# Data Cleaning

- Some variables contain outliers
- Some variables have default values
- Some variables have missing values
  
- `RevolvingUtilizationOfUnsecuredL`
- `NumberOfTime30_59DaysPastDueNotW`
- Monthly income has missing values
  
- Shall we delete them and go ahead with our analysis?



# Missing values & Outliers

- Data is not always available E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
  - Equipment malfunction
  - Inconsistent with other recorded data and thus deleted
  - Data not entered due to misunderstanding
  - Certain data may not be considered important at the time of entry
  - Not register history or changes of the data
- Missing data may need to be inferred.
- Missing data - values, attributes, entire records, entire sections
- Missing values and defaults are indistinguishable



# Imputation

---

# Missing Value Imputation1

- **Standalone imputation**

- Mean, median, other point estimates
- Convenient, easy to implement
- **Assume:** Distribution of the missing values is the same as the non-missing values.
- Does not take into account inter-relationships
- **Eg:** The average of available values is 11.4. Can we replace the missing value in this table by **11.4** ?

X1
11.0
11.1
11.9
10.9
10.8
.
11.5
11.6
11.6
11.4
11
12
11.8
11.4
11.9

# Missing Value Imputation2

- Use attribute relationships
- Better imputation
- Two techniques
  - Propensity score (nonparametric). Useful for discrete variables
  - Regression (parametric)
- There are two missing values in x2. What are the most appropriate replacements

X1	X2
-4	-12
2	6
-6	-18
8	24
-1	
-4	-12
-5	-15
4	12
-4	-12
-5	-15
-2	
4	12
10	30
-10	-30
-3	-9

# Missing Value Imputation3

- There are two missing values in x2. Find the most appropriate replacements

x1	x2
4	1
5	1
4	1
3	1
3	1
4	1
5	1
3	
31	0
39	0
32	0
37	0
32	0
32	0
32	

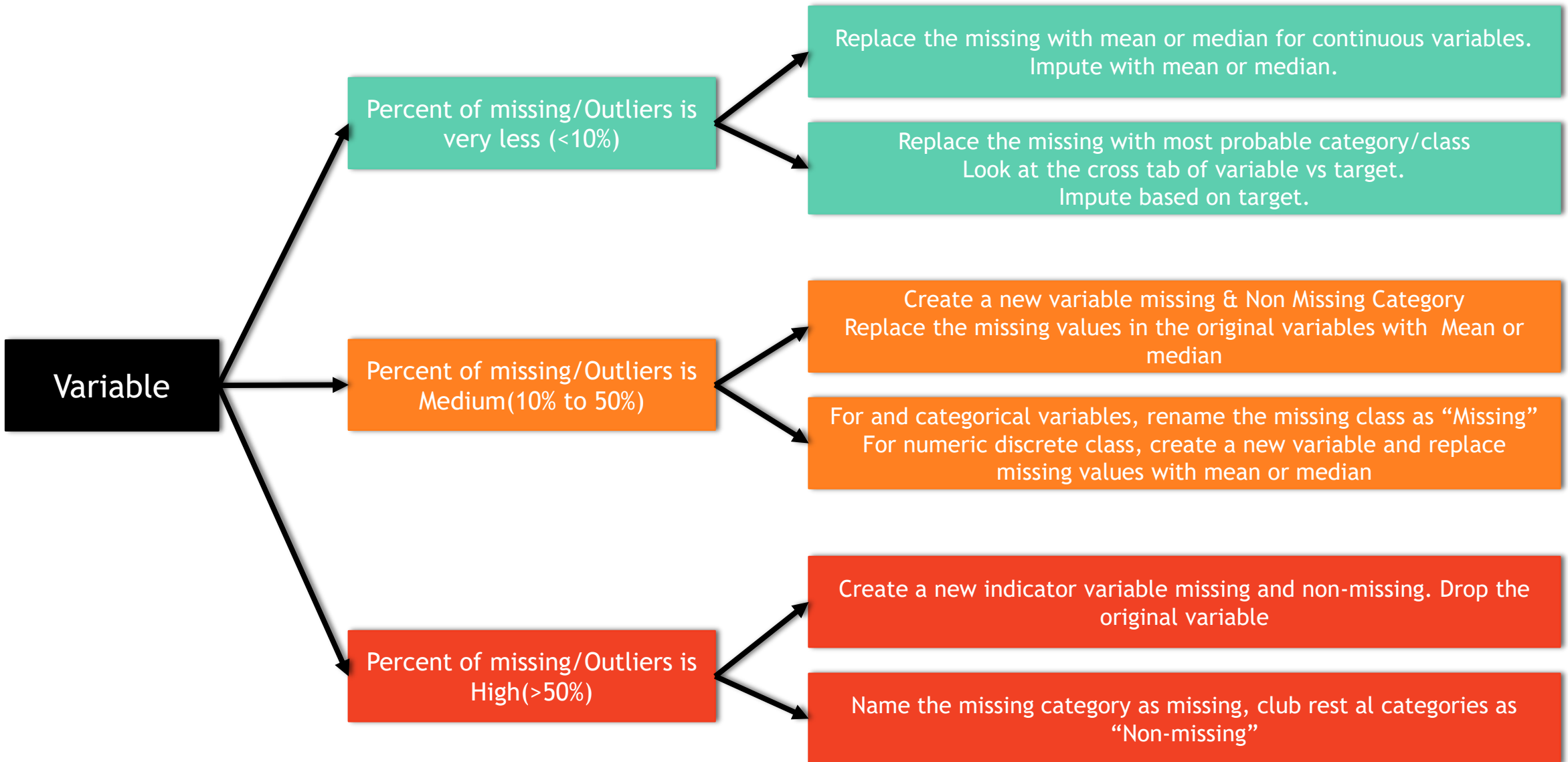
# Missing Value Imputation4

- What if more than 50% are missing?
- It doesn't make sense to carry out the analysis on 20% or 30% of the whole data and give inferences on overall data
- The best imputation is ignore the actual values and take available or not available info



# Step-4: Missing Values and Outlier Treatment

---





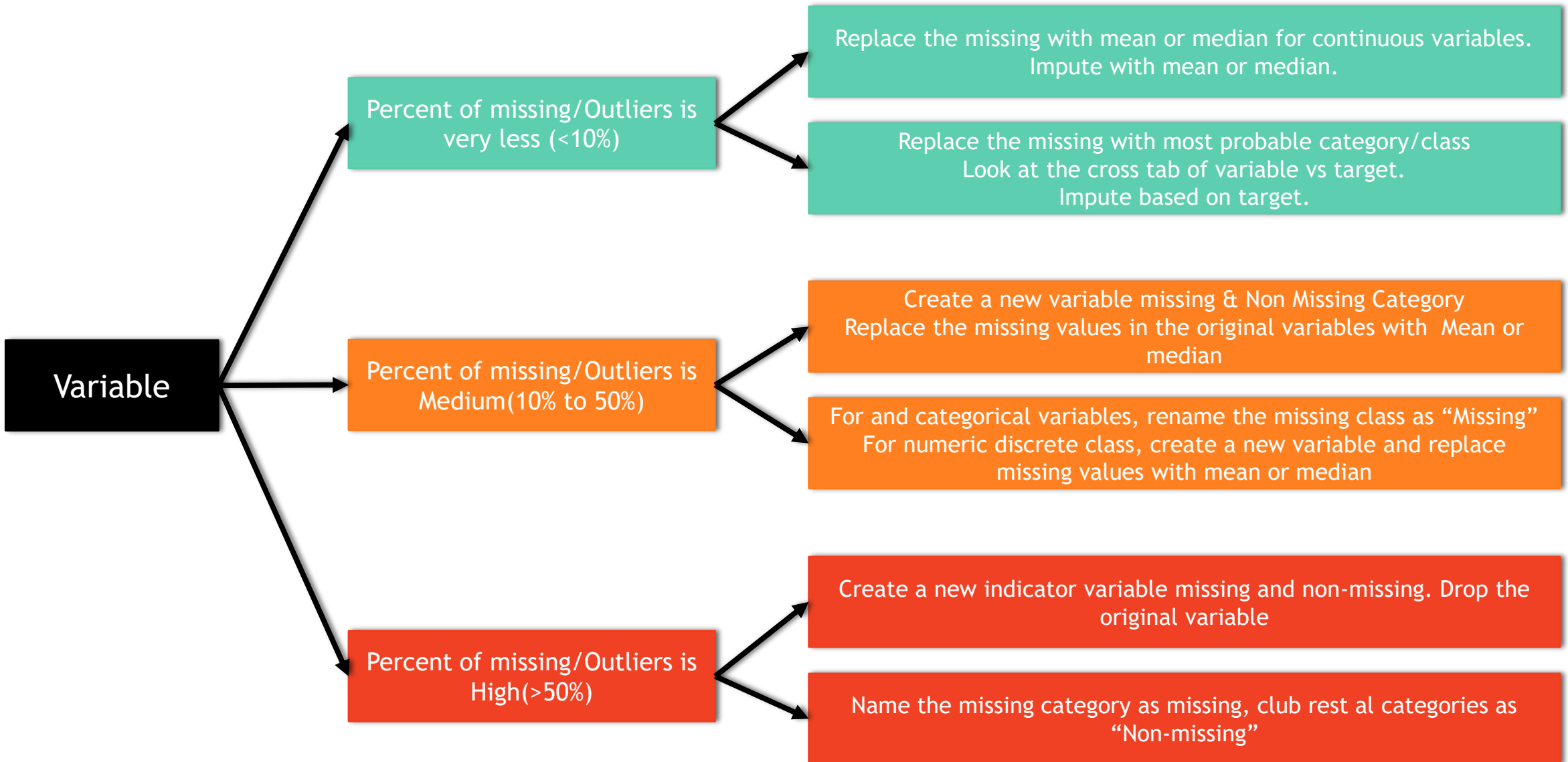


# Data Cleaning Scenario-1

---

# RevolvingUtilizationOfUnsecuredLines

- RevolvingUtilizationOfUnsecuredLines has outliers.
- What type of variable is this? What are the possible values?
- Its' mean is 6.05 which is greater than 1. So variable has some faulty values. Its maximum value is 50710 which is way too high.
- Lets look at percentiles to know from where it is exceeding 1.



Variable

Percent of missing/Outliers is  
very less (<10%)

Replace the missing with mean or median for continuous variables.  
Impute with mean or median.

# Data Cleaning

- RevolvingUtilizationOfUnsecuredLines has outliers.
- Since outliers percentage is less than 10% We will replace outliers with mean of remaining data.
- Outliers are with value greater than 1.

# LAB: Data Cleaning Scenario-1

- What percent are missing values in RevolvingUtilizationOfUnsecuredLines?
- Get the detailed percentile distribution
- Clean the variable, and create a new variable by removing all the issues

# Code: Data Cleaning Scenario-1

```
# # LAB: Data Cleaning Scenario-1
#What percent are missing values in
RevolvingUtilizationOfUnsecuredLines?
#Get the detailed percentile distribution
util_percentiles=loans['RevolvingUtilizationOfUnsecuredLines'].q
uantile([0.05, 0.1, 0.25, 0.5, 0.75, 0.80,
0.9,0.91,0.95,0.96,0.97,0.975,0.98,0.99,1])
round(util_percentiles,2)

#Clean the variable, and create a new variable by removing all
the issues
#If utilization is more than 1 then it can be replaced by median
median_util=loans['RevolvingUtilizationOfUnsecuredLines'].median
()
median_util
```

# Code: Data Cleaning Scenario-1

```
loans['util_new']=loans['RevolvingUtilizationOfUnsecuredLines']
loans['util_new'][loans['util_new']>1]=median_util
loans['util_new']

# percentile distribution for new variable
util_percentiles1=loans['util_new'].quantile([0.05, 0.1, 0.25, 0.5, 0.75, 0.80,
0.9,0.91,0.95,0.96,0.97,0.975,0.98,0.99,1])
round(util_percentiles1,2)
```



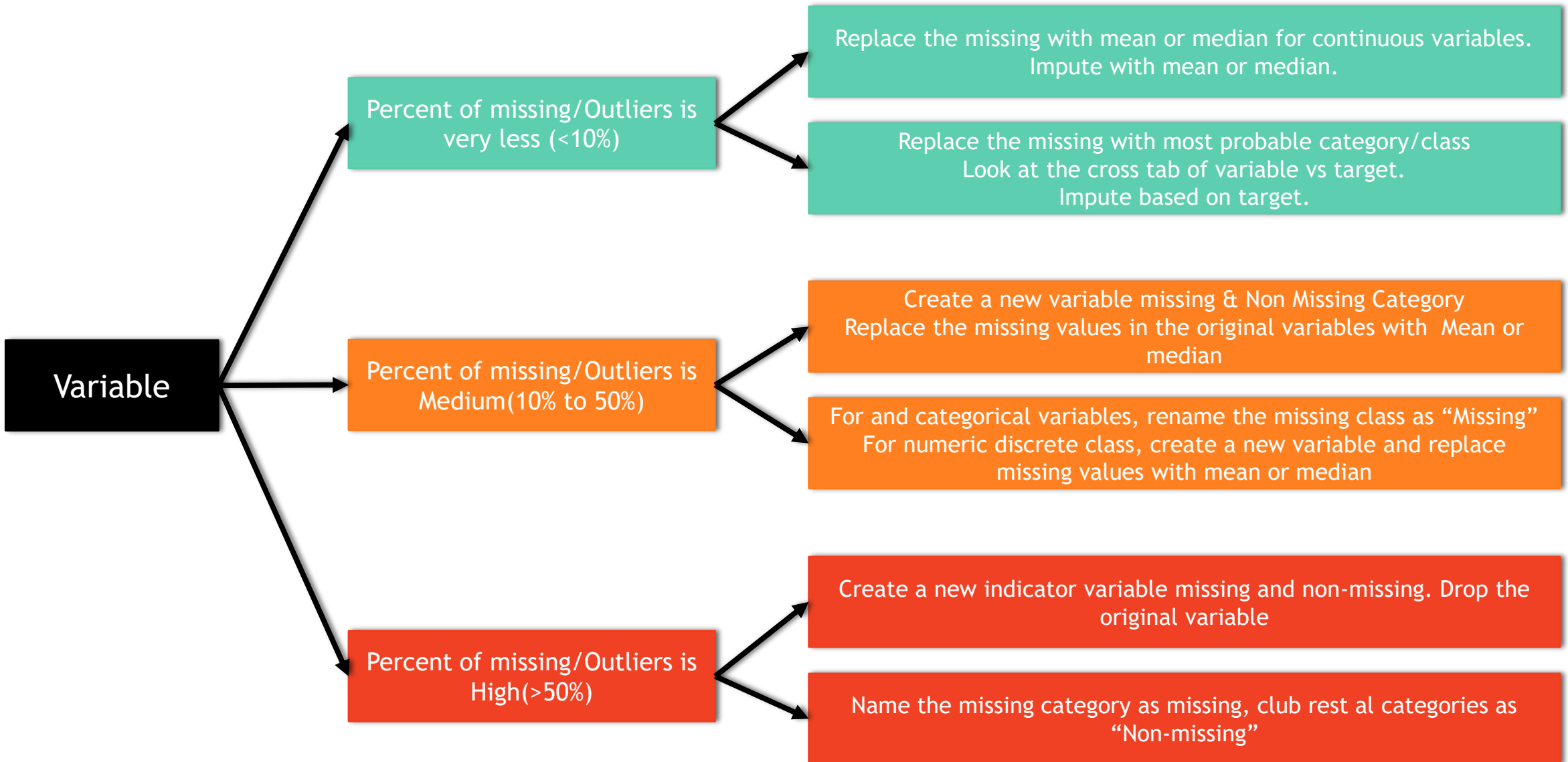


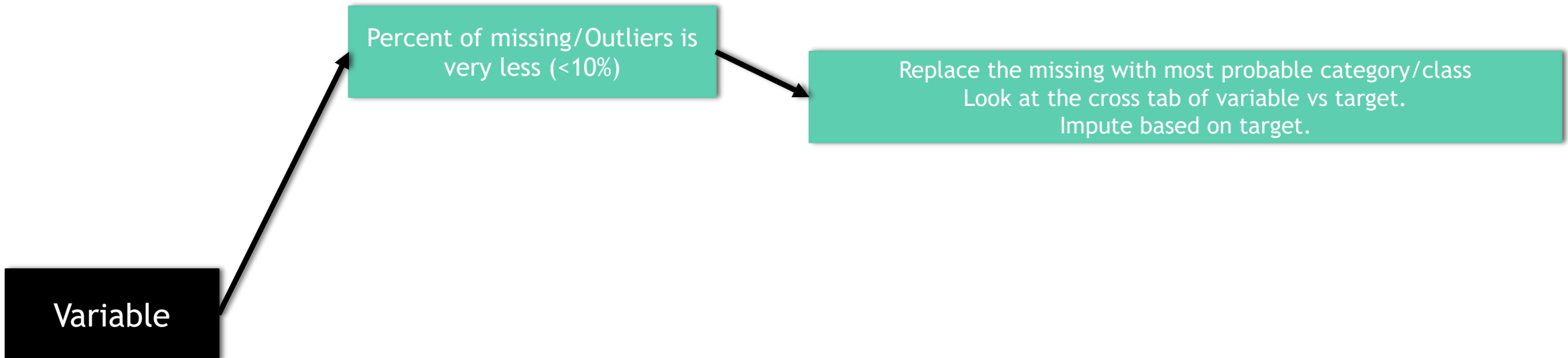
# Data Cleaning Scenario-2

---

# NumberOfTime30\_59DaysPastDueNotW

- Find bad rate in each category of this variable
- Replace 96 with \_\_\_\_? Replace 98 with \_\_\_\_?





# LAB: Data Cleaning Scenario-2

- What is the issue with NumberOfTime30\_59DaysPastDueNotW
- Draw a frequency table
- What percent of the values are erroneous?
- Clean the variable- Look at the cross tab of variable vs target. Impute based on target .
- Create frequency table for cleaned variable

# Code: Data Cleaning Scenario-2

```
# # LAB: Data Cleaning Scenario-2
#What is the issue with NumberOfTime30_59DaysPastDueNotW
#Draw a frequency table
freq_table_30dpd=loans['NumberOfTime30-59DaysPastDueNotWorse'].value_counts(sort=False)
freq_table_30dpd

#One month defaults frequency can't be beyond 24 in last 24 months
#What percent of the values are erroneous?
freq_table_30dpd[13:len(freq_table_30dpd)]
freq_table_30dpd[13:len(freq_table_30dpd)].sum()/freq_table_30dpd.sum()

#Clean the variable- Look at the cross tab of variable vs target. Impute based on target .
#Cross tab with target
import pandas as pd
cross_tab_30dpd_target=pd.crosstab(loans['NumberOfTime30-
59DaysPastDueNotWorse'],loans['SeriousDlqin2yrs'])
cross_tab_30dpd_target
```

# Code: Data Cleaning Scenario-2

```
#Cross tab row Percentages
cross_tab_30dpd_target_percent=cross_tab_30dpd_target.astype(float).div(cross_tab
_30dpd_target.sum(axis=1), axis=0)
round(cross_tab_30dpd_target_percent,2)

loans['num_30_59_dpd_new']=loans['NumberOfTime30-59DaysPastDueNotWorse']
loans['num_30_59_dpd_new'][loans['num_30_59_dpd_new']>12]=6
loans['num_30_59_dpd_new']

loans['num_30_59_dpd_new'].value_counts(sort=False)
```



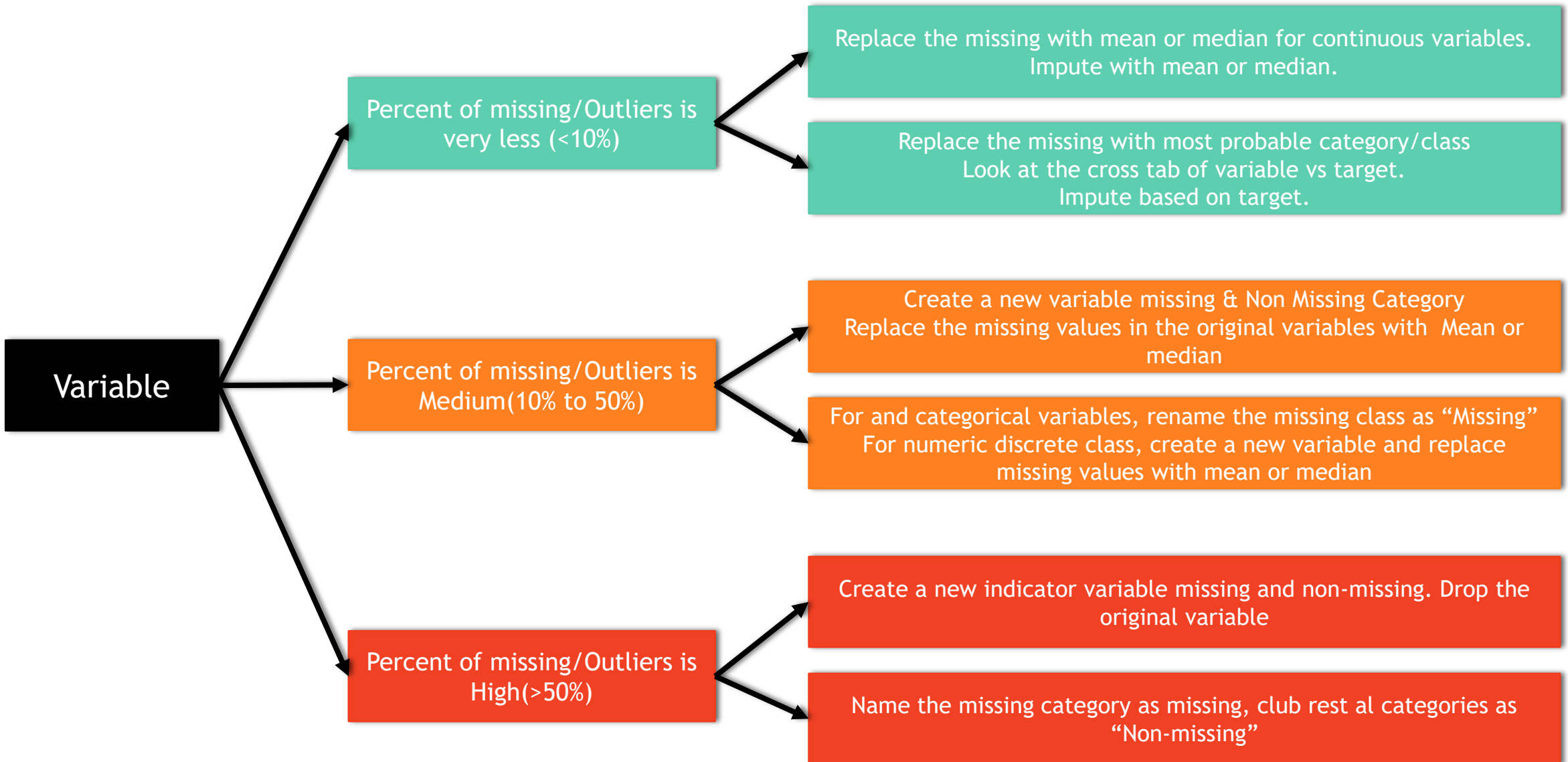
# Data Cleaning Scenario-3

---



# Monthly Income

- Monthly Income has nearly 20% missing values
- Missing value percentage is significant
- Simply replacing with mean or median is not sufficient
- We can create an indicator variable to keep track of missing and non-missing values



Variable

Percent of missing/Outliers is  
Medium(10% to 50%)

Create a new variable missing & Non Missing Category  
Replace the missing values in the original variables with Mean or  
median

# LAB: Monthly Income

- Find the missing value percentage in monthly income
- Create an indicator variable for missing and non-missing
- Replace the missing values with median

# LAB: Monthly Income

```
# # Data Cleaning Scenario-3
#Find the missing value percentage in monthly income
loans['MonthlyIncome'].isnull().sum()
loans['MonthlyIncome'].isnull().sum()/len(loans)
#Once identified where missing values exist, the next task usually is to fill
them (data imputation). Depending upon the context,
#in this case, I am assigning median value to all those positions where missing
value is present:

loans['MonthlyIncome_ind']=1
loans['MonthlyIncome_ind'][loans['MonthlyIncome'].isnull()]=0
loans['MonthlyIncome_ind'].value_counts(sort=False)
```

# LAB: Monthly Income

```
loans['MonthlyIncome_new']=loans['MonthlyIncome']  
loans['MonthlyIncome_new'][loans['MonthlyIncome'].isnull()]=loans['MonthlyIncome'].median()  
round(loans['MonthlyIncome_new'].describe())
```



# Data Cleaning Other Variables

---

# Remaining Variables Imputation

- Debt Ratio: Imputation like monthly income
- NumberOfOpenCreditLinesAndLoans : Imputation
- NumberOfTimes90DaysLate: Imputation similar to NumberOfTime30\_59DaysPastDueNotW
- NumberRealEstateLoansOrLines: : Imputation
- NumberOfTime60\_89DaysPastDueNotW: Imputation similar to NumberOfTime30\_59DaysPastDueNotW
- NumberOfDependents: Impute based on target variable





# Conclusion

---

# Conclusion

- Data cleaning is as important as data analysis
- Sometimes 80% of the overall project time is spent on data cleaning
- Data cleaning needs patience, we need to clean for each individual variable
- Apart from suggested methods, there are many heuristic ways of cleaning the data