A picture containing shape, arrow

Description automatically generated

**Imputation**

Instructions:

Please share your answers filled inline in the word document. Submit code files wherever applicable.

Please ensure you update all the details:

**Name: Gunjan Kumar Gupta**

**Batch Id: DSWDMCSR 300522B**

**Topic: Data Pre-Processing**

**Problem Statement:**

Majority of the datasets have missing values, that might be because the data collected were not at regular intervals or the breakdown of instruments and so on. It is nearly impossible to build the proper model or in other words, get accurate results. The common techniques are either removing those records completely or substitute those missing values with the logical ones, there are various techniques to treat these types of problems.

1. Prepare the dataset using various techniques to solve the problem, explore all the techniques available and use them to see which gives the best result.

**Hint:**  Go through this link: <https://360digitmg.com/mindmap-data-science>

**A picture containing shape, arrow

Description automatically generated**

**Hints:**

For each assignment, the solution should be submitted in the below format

1. Work on every feature of the dataset and create a data dictionary as an example displayed in the image below:



1. Hint: Refer to the file Claimants.csv.
2. The data is a vehicle Insurance data. Research on the Data fields and perform preliminary analysis
3. Research and perform all possible steps for obtaining solution
4. All the codes (executable programs) should execute without errors
5. Code modularization should be followed
6. Each line of code should have comments explaining the logic and why you are using that function

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_missing value \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_##

import numpy as np

import pandas as pd

# load data

df\_m = pd.read\_csv(r"D:\DATA Science 360 DigiTMG\Assignment\Data Preprocessing\_Assignments\DataSets-Data Pre Processing\claimants.csv")

df\_m.dtypes

df\_m.columns

# check for count of NA's in each column

df\_m.isna().sum()

# for Mean, Meadian, Mode imputation we can use Simple Imputer or df.fillna()

from sklearn.impute import SimpleImputer

# Mean Imputer CLMSEX

mean\_imputer = SimpleImputer(missing\_values=np.nan, strategy='mean')

df\_m["CLMSEX"] = pd.DataFrame(mean\_imputer.fit\_transform(df\_m[["CLMSEX"]]))

df\_m["CLMSEX"].isna().sum()

# df\_m.isna().sum()

# Mean Imputer CLMINSUR

mean\_imputer = SimpleImputer(missing\_values=np.nan, strategy='mean')

df\_m["CLMINSUR"] = pd.DataFrame(mean\_imputer.fit\_transform(df\_m[["CLMINSUR"]]))

df\_m["CLMINSUR"].isna().sum()

# Mean Imputer SEATBELT

mean\_imputer = SimpleImputer(missing\_values=np.nan, strategy='mean')

df\_m["SEATBELT"] = pd.DataFrame(mean\_imputer.fit\_transform(df\_m[["SEATBELT"]]))

df\_m["SEATBELT"].isna().sum()

# Mean Imputer CLMAGE

mean\_imputer = SimpleImputer(missing\_values=np.nan, strategy='mean')

df\_m["CLMAGE"] = pd.DataFrame(mean\_imputer.fit\_transform(df\_m[["CLMAGE"]]))

df\_m["CLMAGE"].isna().sum()

df\_m.isna().sum() # No Missing Value