A picture containing shape, arrow

Description automatically generated

**Imputation**

Instructions:

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

Please ensure you update all the details:

**Name: ANANADAKRISHNAN K V**

**Batch Id: 190421**

**Topic: Preliminaries for Data Analysis**

**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**

**Ans:-**

**Python code**

################ imputation ###################

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from scipy import stats

from sklearn.impute import SimpleImputer as si

### import data set claimants as data frame

a\_main = pd.DataFrame(claimantscsv)

## creating copy thereby keep original data safe

d = a\_main.copy(deep=True)

### CASENUM, ATTORNY AND LOSS column have ot any missing values. so we no need to go for

# any imputation there.

### mean imputater ###

d = a\_main.copy(deep=True)

a= d.copy(deep=True)

a.isna()

print(a.isna().sum())

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

a.iloc[:,:] = mean\_imputer.fit\_transform(a)

print(a.isna().sum())

### median imputer ###

d = a\_main.copy(deep=True)

b= d.copy(deep=True)

b.isna()

print(b.isna().sum())

median\_imputer = si(missing\_values=np.nan, strategy='median')

b.iloc[:,:] = median\_imputer.fit\_transform(b)

print(b.isna().sum())

### mode imputer ###

d = a\_main.copy(deep=True)

c= d.copy(deep=True)

c.isna()

print(c.isna().sum())

most\_frequent\_imputer = si(missing\_values=np.nan, strategy='most\_frequent')

c.iloc[:,:] = most\_frequent\_imputer.fit\_transform(c)

print(c.isna().sum())

### OUTLIERS ###

## no.of outliers in CASENUM = 57

## no.of outliers in CLMAGE = 1

## no.of outliers in LOSS = 66

### conclussion:-

# since CASENUM, ATTORNY & LOSS have not any missing values we are gonna completly eliminate those columns to do any imputation.

# Here CLMAGE is continuous data,so we can apply mean & median imputation there. CLMAGE column have only one outlier. So its mean value

# should almost near to actual centrality. so we are gonna apply mean imputation for CLMAGE column. and mode imputation for

# CLMSEX, CLMINSUR & SEATBELT

### BEST IMPUTATION STRATEGY ###

# mean imputation on continuous data

d = a\_main.copy(deep=True)

d["CLMAGE"]

d.CLMAGE.isna()

print(d.CLMAGE.isna().sum())

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

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

print(d.CLMAGE.isna().sum())

## mode imputation on categorical data

d["CLMSEX"]

d["CLMSEX"].isna()

print(d["CLMSEX"].isna().sum())

most\_frequent\_imputer = si(missing\_values=np.nan, strategy='most\_frequent')

d["CLMSEX"] = pd.DataFrame(most\_frequent\_imputer.fit\_transform(d[["CLMSEX"]]))

print(d.CLMSEX.isna().sum())

d["CLMINSUR"]

d["CLMINSUR"].isna()

print(d["CLMINSUR"].isna().sum())

most\_frequent\_imputer = si(missing\_values=np.nan, strategy='most\_frequent')

d["CLMINSUR"] = pd.DataFrame(most\_frequent\_imputer.fit\_transform(d[["CLMINSUR"]]))

print(d.CLMINSUR.isna().sum())

d["SEATBELT"]

d["SEATBELT"].isna()

print(d["SEATBELT"].isna().sum())

most\_frequent\_imputer = si(missing\_values=np.nan, strategy='most\_frequent')

d["SEATBELT"] = pd.DataFrame(most\_frequent\_imputer.fit\_transform(d[["SEATBELT"]]))

print(d.SEATBELT.isna().sum())

# RESULT

print("best imputed data is",d)

print(d.isna().sum())

**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