Dt:- 21/12/22
5:50pm

* Data Preprocessing / Data Preparation

⇒ Feature Engineering

Data cleaning
- Missing Values
- Outliers

+

Data Wrangling
↳ (Data Transformation)
- Encoding
- Scaling
- Transformation.

Us.
(Car Insurance)

Data cleaning (In spyder)

```
# import numpy as np
# import pandas as pd
```

Why two Dataset
Because to check old(
and new dataset.

⇒ Load Data set

```
df_raw = pd.read_csv("claimants.csv")
# df = pd.read_csv("claimants.csv")
```

Out:- df_raw | dataFrame | 1340,7 | column name : CASENUM,
df | dataframe | 1340,7 | column names : "

⇒ Check is there null values.

```
# df.isnull().sum()
```

Out :-

CASENUM	0
CLMSEX	12
CLMINSUR	41
SEATBELT	48
CLMAGE	189
LOSS	0
ATTORNEY	0

MISSING Values.

Data Understanding

Case Num :- Number of case (insurance)

Discrete clm sex :- 0, 1 [Male] [Female]

Discrete clm insur :- 1(Yes), 0 [NO] # having Insurance or not

Discrete Seat belt :- 0 [NO], 1 [Yes] # wearing seat belt

Continuoy clm age :- Age of the person

Loss :- Loss / Damage happened

attorney :- Whether they hire a lawyer (or) not, [0] No, [1] Yes.

Person who is Claming Isurance is called "claimants" During the accident

There are various methods for Filling null values

* For Continous Data
  - mean (no outliers)
  - median (outliers)
* For Discrete Data → Mode

Because Mean is Heavily Impacted with Outliers [Less affected]

Replacing null values For discrete Variables

So, we have to Replace Them with "MODE"

```
# df["CLMSEX"].value_counts()
```

(MODE)

Out :- 1.0 742 → Highest, so we Replace with "1"
0.0 586

```
# df["CLMSEX"].fillna(1, inplace=True)
```

↓ changing in orginal DaTa = "TRUE"

Out Filled with "1"

```
# df["CLMINSUR"].value_counts()
```

Out :- 1.0 1220 → MODE
0.0 120

```
# df["CLMINSUR"].fillna(1, inplace=True)
```

Out Filled with "1", so, replaced with "1" value

df["SEATBELT"].valuecounts()

Out :- 0.0 1270 → Mode.
1.0 22

df["SEATBELT"].fillna(0, inplace = True)

Out :- Filled with "0"

Time 11:30pm

Replacing null Values For Continous Variables By #Pandas

df["CLMAGE"].mean()

Out 28.414

df["CLMAGE"].median()

Out 30.00

df["CLMAGE"].fillna(28.414, inplace True)

Out Filled with 28.414

If we ask again what is median value.

df["CLMAGE"].median()

Out 28.414 , so it changes 30.00 To 28.414
Because we Filled with null values 28.414.

2nd method

Sklearn (Scikit Learn) (One stop shop For ML)

* SimpleImputer

Load data set again

Liberary → (module) (Function)

```
From Sklearn.impute import SimpleImputer
```

Ex: missing values = "???"

Calculating mean value

```
# mean_imputer = SimpleImputer (strategy = "mean") → E/n
```

(Function) → (Argument)

name as

Function. (or) Store in mean_imputer

Ex:- Core Python [Keyword argument]

If we don't write this line it gives

Same : a = 4

Converting array To data Frame

To Remove array. From mean value.

```
# df ["CLMAGE"] = Pd.DataFrame (mean_imputer.fit_trans
```

storing

```
_Trans Form (df [["CLMAGE"]]))
```

```
array ([50.],
       [18.],
       [5.],
       ⋮
```

Fill in This column (or) Replacing

Calculate The mean value

Out

Filled with 28.414

For Median only change mean To median

```
# median_imputer = SimpleImputer (strategy = "Median")
```

```
# df["CLMAGE"] = pd.DataFrame(median_imputer.fit_
                Transform(df[["CLMAGE"]])
```

For Mode [Discrete Data]

```
mode_imputer = SimpleImputer(strategy="mode")

#df["CLMSEX"] = pd.DataFrame(mode_imputer.fit_transform(df[[
                                    "CLMSEX"]])

# df["CLMINSUR"] = pd.DataFrame(mode_imputer.fit_transfo
                        (df[["CLMINSUR"]])

#df("SEATBELT"] = pd.Data Frame( mode_imputer.fit_trans
                        m(df[["SEATBELT"]])
```

Output — Fills Directly The mode Value in to the dataset.

MODE = it is for Discrete DaTa.

MEAN
MEDIAN
} → Continous DaTa