What is Data Preprocessing

- Data Contain text,image,video,time,audio.
- When we take raw data there will be some problem or error so to overcome these we perform data preprocess.
- It is a process that convert raw data into meaningful data using differnet techinque.
- Data In real world may consit of incomplete,noise,incosistant,duplicate it affect our model accuracy

Different Step For Data Preprocessing

```
 1. Data Cleaning - It means fill missing value,smooth out the noise while identif
y the outlier, and correct the data.
 2. Data Integration - Here we merge data from differnet source in to a coherent d
ata store such as data warehouse.
 3. Data Reduction -It is a technique in which we can reduce the data size by aggr
egating,elimanating redundant feature.
 4. Data Transformation - Data are transformed into appropriate forms for ml mode
l.here we use normalization technique.
 5. Data Discretization - in this techinque transforms numeric data by mapping val
ues to interval or concept labels.
```

1. Handling Missing Value

- We can classify the missing values in different types. Each type of missing value require slightly different handling. The main types are —

```
1. Missing completely at Random (MCAR) : As the name suggests missing completely a
t random means that there’s no relationship between whether a data point is missin
g and any values in the data set, missing or observed. The missing data is just a
 random subset of the data.


2. Missing at Random (MAR) : Missing at random means that the propensity of missin
g values has a systematic relationship with the observed data but not the missing
 data.


3. Missing Not at Random (MNAR) : Missing not at random means that there is a dist
inct relationship between the propensity of a value to be missing and its values.
```

Techniques of dealing with missing data

```
1. Drop missing values/columns/rows
2. Imputation
```

Dropping Missing data

- The simplest way to drop the columns/rows for which the data is not available.
- In the first two types of missing data, MCAR and MAR, in general, it is safe to remove the data with missing values depending upon their occurrences.
- but in the third case (MNAR) removing observations with missing values can produce a bias in the model.

- Lets take a dataset and see how to handle missing values in it.

In [1]:

```
import pandas as pd
import numpy as np
data = pd.read_csv('Melbourne_housing_FULL.csv')
data.head(5)
```

Out[1]:

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postc
0	Abbotsford	68 Studley St	2	h	NaN	SS	Jellis	3/09/2016	2.5	30
1	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016	2.5	30
2	Abbotsford	25 Bloomburg St	2	h	1035000.0	S	Biggin	4/02/2016	2.5	30
3	Abbotsford	18/659 Victoria St	3	u	NaN	VB	Rounds	4/02/2016	2.5	30
4	Abbotsford	5 Charles St	3	h	1465000.0	SP	Biggin	4/03/2017	2.5	30

5 rows × 21 columns

Now 1st step to check wheather this data set contain missing value or not.

In [2]:

```
data_temp = data.copy()
```

In [3]:

```
data_temp.isnull().sum()
```

Out[3]:

```
Suburb              0
Address             0
Rooms               0
Type                0
Price            7610
Method              0
SellerG             0
Date                0
Distance            1
Postcode            1
Bedroom2         8217
Bathroom         8226
Car              8728
Landsize        11810
BuildingArea    21115
YearBuilt       19306
CouncilArea         3
Lattitude        7976
Longtitude       7976
Regionname          3
Propertycount       3
dtype: int64
```

Now you can see there are huge number of missing value now apply 1st method that drop the missing value

In [4]:

```
#dropna function in Pandas removes all the rows with missing values
data_temp.dropna(inplace=True)
#Putting axis=1 removes the columns with missing values
data_temp.dropna(inplace=True, axis=0)
```

In [5]:

```
data_temp.isnull().sum()
```

Out[5]:

```
Suburb           0
Address          0
Rooms            0
Type             0
Price            0
Method           0
SellerG          0
Date             0
Distance         0
Postcode         0
Bedroom2         0
Bathroom         0
Car              0
Landsize         0
BuildingArea     0
YearBuilt        0
CouncilArea      0
Lattitude        0
Longtitude       0
Regionname       0
Propertycount    0
dtype: int64
```

In [6]:

```
print("Actual data :",data.shape)
print("after drop na value :",data_temp.shape)
```

```
Actual data : (34857, 21)
after drop na value : (8887, 21)
```

- If we drop the rows our total number of data points to train our model will go down which can reduce the model performance.
- Do this only if you have large number of training examples and the rows with missing data are not very high in number. Dropping the column altogether will remove a feature from our model i.e the model predictions will be independent of the building area.
- So dropping the misssing value is not good for any model as the number of data reduce so Lets look at a better approach for dealing with missing data.

Imputation

- Imputation means to replace or fill the missing data with some value.
- There are lot of ways to impute the data.

    1. A constant value that belongs to the set of possible values of that variable, such as 0, distinct from all other values
    2. A mean, median or mode value for the column

Lets go back to our dataset and see how each of these methods work

1st approach we can just put zero value in place of missing value.

impute with 0

In [7]:

```
data_temp2 = data.copy()
```

In [8]:

```
data_temp2.isnull().sum()
```

Out[8]:

```
Suburb              0
Address             0
Rooms               0
Type                0
Price            7610
Method              0
SellerG             0
Date                0
Distance            1
Postcode            1
Bedroom2         8217
Bathroom         8226
Car              8728
Landsize        11810
BuildingArea    21115
YearBuilt       19306
CouncilArea         3
Lattitude        7976
Longtitude       7976
Regionname          3
Propertycount       3
dtype: int64
```

In [9]:

```
data_temp2.fillna(0, inplace=True)
```

see here all missing value replaced by zero. but it also not a good approach for better model so now we see how we impute mean ,median and mode

In [10]:

```
data_temp2.isnull().sum()
```

Out[10]:

```
Suburb           0
Address          0
Rooms            0
Type             0
Price            0
Method           0
SellerG          0
Date             0
Distance         0
Postcode         0
Bedroom2         0
Bathroom         0
Car              0
Landsize         0
BuildingArea     0
YearBuilt        0
CouncilArea      0
Lattitude        0
Longtitude       0
Regionname       0
Propertycount    0
dtype: int64
```

Use Mean,Median,Mode

In [11]:

```
data_temp3 = data.copy()
```

In [12]:

```
data_temp3.isnull().sum()
```

Out[12]:

```
Suburb              0
Address             0
Rooms               0
Type                0
Price            7610
Method              0
SellerG             0
Date                0
Distance            1
Postcode            1
Bedroom2         8217
Bathroom         8226
Car              8728
Landsize        11810
BuildingArea    21115
YearBuilt       19306
CouncilArea         3
Lattitude        7976
Longtitude       7976
Regionname          3
Propertycount       3
dtype: int64
```

Imp Point to remember

- we all know basically three type of data int,float and object.
- so when we impute missing value in (int and float) we can use all mean,median and mode, but you can always use mean and median, but if the data set contain outlier then you definitly use median as mean affect by outlier. so better in int and float you must use meadin.
- Now in object you need to use mode only other wise it throws error. when you use mode keep in mind you should use [0] after mode so that the exact data can be extract.

Summerize

- int,float - use Mean(), Median()
- object - mode()[0]

Here you impute one by one means in int,float you impute mean,median and in object mode.

In [13]:

```
# object data i use mode()[0]
print("Missing Value before use imputation : ",data_temp3.CouncilArea.isnull().sum().sum())

data_temp3['CouncilArea'] = data_temp3['CouncilArea'].fillna(data_temp3['CouncilArea'].mode

print("Missing Value after use imputation : ",data_temp3.CouncilArea.isnull().sum().sum())
```

```
Missing Value before use imputation :  3
Missing Value after use imputation :  0
```

In [14]:

```
# float data i use median()
print("Missing Value before use imputation : ",data_temp3.Longtitude.isnull().sum().sum())

data_temp3['Longtitude'] = data_temp3['Longtitude'].fillna(data_temp3['Longtitude'].median(

print("Missing Value before use imputation : ",data_temp3.Longtitude.isnull().sum().sum())
```

```
Missing Value before use imputation :  7976
Missing Value before use imputation :  0
```

in this way you can fixed missing value in your data. its just basic technique there are other also available