1. Introduction

We all know the Importance of good features for machine learning models. In Machine learning task we have features which we need to process to make them good and this is done by data preprocessing tasks.

Data Preprocessing : Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

Data Preprocessing involves below steps:

- Creating the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data

Many of us know traditional approaches for above listed steps but in this notebook I will discuss some different approaches which could be game changer for your next project.

In [1]:

```python
# Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```python
# Create dataset from pandas Series
pop = pd.Series({'Karnataka':61095297, 'Tamil Nadu':72147030, 'Andhra Pradesh':None, 'Kerala':33406061})
area = pd.Series({'Karnataka':191791, 'Tamil Nadu':130060, 'Andhra Pradesh':162968, 'Kerala':None})
states = pd.DataFrame({'Population':pop,'Area':area})
states
```

Out[2]:

	Population	Area
Karnataka	61095297.0	191791.0
Tamil Nadu	72147030.0	130060.0
Andhra Pradesh	NaN	162968.0
Kerala	33406061.0	NaN

Understanding the data

In [3]:

```
states.columns
```

Out[3]:

```
Index(['Population', 'Area'], dtype='object')
```

In [4]:

```
states.index
```

Out[4]:

```
Index(['Karnataka', 'Tamil Nadu', 'Andhra Pradesh', 'Kerala'], dtype='object')
```

In [5]:

```
states.count()
```

Out[5]:

```
Population    3
Area          3
dtype: int64
```

In [6]:

```
states.sum()
```

Out[6]:

```
Population    166648388.0
Area             484819.0
dtype: float64
```

In [7]:

```
states.mean()
```

Out[7]:

```
Population    5.554946e+07
Area          1.616063e+05
dtype: float64
```

In [8]:

```
states.median()
```

Out[8]:

```
Population    61095297.0
Area            162968.0
dtype: float64
```

In [9]:

```
states.mode()
```

Out[9]:

	Population	Area
0	33406061.0	130060.0
1	61095297.0	162968.0
2	72147030.0	191791.0

In [10]:

```
states.std()
```

Out[10]:

```
Population    1.995703e+07
Area          3.088802e+04
dtype: float64
```

In [11]:

```
states.min()
```

Out[11]:

```
Population    33406061.0
Area            130060.0
dtype: float64
```

In [12]:

```
states.max()
```

Out[12]:

```
Population    72147030.0
Area            191791.0
dtype: float64
```

In [13]:

```
states.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 4 entries, Karnataka to Kerala
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Population  3 non-null      float64
 1   Area        3 non-null      float64
dtypes: float64(2)
memory usage: 96.0+ bytes
```

In [14]:

```
states.describe()
```

Out[14]:

	Population	Area
count	3.000000e+00	3.000000
mean	5.554946e+07	161606.333333
std	1.995703e+07	30888.018589
min	3.340606e+07	130060.000000
25%	4.725068e+07	146514.000000
50%	6.109530e+07	162968.000000
75%	6.662116e+07	177379.500000
max	7.214703e+07	191791.000000

2. Handling Missing Values

Missing data are not rare in real data sets. In fact, the chance that at least one data point is missing increases as the data set size increases. Missing data can occur any number of ways, some of which include the following.

- Merging of source data sets
- Random events
- Failures of measurement

In [15]:

```
# Handling Missing Values
states.isnull()
```

Out[15]:

	Population	Area
Karnataka	False	False
Tamil Nadu	False	False
Andhra Pradesh	True	False
Kerala	False	True

In [16]:

```
# Columns having missing values
missing_columns = [col for col in states.columns if states[col].isnull().sum() > 0]
missing_columns
```

Out[16]:

```
['Population', 'Area']
```

In [17]:

```
states.isnull().any()
```

Out[17]:

```
Population    True
Area          True
dtype: bool
```

In [18]:

```
states.isnull().sum()
```

Out[18]:

```
Population    1
Area          1
dtype: int64
```

2.2 Methods to Handle Missing Data

As we Know if our data has missing values than our model will not train except few models which can tolerate them like some tree based models but the point is we want to handle this and how can we handle them. So, in this notebook to handle missing data I will discuss following techniques :-

- Deletion of Data
- Encoding Missingness
- Imputation Methods

2.2.1 Deletion of Data

The simplest approach for dealing with missing values is to remove entire attribute(s) and/or sample(s) that contain missing values. However, one must carefully consider a number of aspects of the data prior to taking this approach. For example, missing values could be eliminated by removing all predictors that contain at least one missing value. Similarly, missing values could be eliminated by removing all samples with any missing values.

Note: When it is difficult to obtain samples or when the data contain a small number of samples (i.e., rows), then it is not desirable to remove samples from the data.

Consider this small intuition shown below

Let M = Number of Samples(rows).\ and Let N = Number of Attributes(columns).

Case 1: Deletion of Attributes

If N has range of [1-10]\ Then don't delete the attribute that contain missing values but if that attribute has missing values around 80-90% then deletion of that attribute will be good option instead of just predicting values of those 80-90% data based on that 10-20% data.

Case 2: Deletion of Samples

If M is a large number according to your task\ Then deletion of sample can be a Good step but if that sample has few missing values with respect to attribute, then you should consider methods to fill those missing values.

Lets move on to the implementation part, I will just show how to delete data for both cases but you can interpret more according to your tasks.

Deletion of an Attribute

According to Simple numerical Summaries the attribute Upfront_charges has largest missing values percentage of (26.664%) which is not ideal percentage to remove a feature but just for sake of implementation I will remove that feature.

In [19]:

```
states.dropna()
```

Out[19]:

	Population	Area
Karnataka	61095297.0	191791.0
Tamil Nadu	72147030.0	130060.0

In [20]:

```
states.fillna(0)
```

Out[20]:

	Population	Area
Karnataka	61095297.0	191791.0
Tamil Nadu	72147030.0	130060.0
Andhra Pradesh	0.0	162968.0
Kerala	33406061.0	0.0

In [21]:

```
states.fillna(method="ffill")
```

Out[21]:

	Population	Area
Karnataka	61095297.0	191791.0
Tamil Nadu	72147030.0	130060.0
Andhra Pradesh	72147030.0	162968.0
Kerala	33406061.0	162968.0

In [23]:

```
states.fillna(method="bfill")
```

Out[23]:

	Population	Area
Karnataka	61095297.0	191791.0
Tamil Nadu	72147030.0	130060.0
Andhra Pradesh	33406061.0	162968.0
Kerala	33406061.0	NaN

In [24]:

```
mean_pop=states['Population'].mean()
mean_area=states['Area'].mean()
states['Population'].fillna(value=mean_pop, inplace=True)
states['Area'].fillna(value=mean_area, inplace=True)
```

In [27]:

```
print(states)
```

```
                  Population           Area
Karnataka       6.109530e+07  191791.000000
Tamil Nadu      7.214703e+07  130060.000000
Andhra Pradesh  5.554946e+07  162968.000000
Kerala          3.340606e+07  161606.333333
```

In [21]:

```
# Save the data into a csv file
states.to_csv('states.csv')
```

Check the path of the current working directory</>

In [31]:

```
pwd
```

Out[31]:

```
'C:\\Users\\Chandrika M\\Documents\\My Subjects\\AI and ML\\Lab'
```

List the files in the current folder

In [34]:

```
ls
```

```
 Volume in drive C is Windows
 Volume Serial Number is 4476-E7F9

 Directory of C:\Users\Chandrika M\Documents\My Subjects\AI and ML\Lab

11-07-2023  09:58    <DIR>          .
11-07-2023  09:35    <DIR>          ..
11-07-2023  09:38    <DIR>          .ipynb_checkpoints
11-07-2023  09:58            33,955 data-preprocessing.ipynb
11-07-2023  09:45               163 states.csv
               2 File(s)         34,118 bytes
               3 Dir(s)  29,889,892,352 bytes free
```

In [24]:

```
df=pd.read_csv('states.csv')
df
```

Out[24]:

	Unnamed: 0	Population	Area
0	Karnataka	6.109530e+07	191791.000000
1	Tamil Nadu	7.214703e+07	130060.000000
2	Andhra Pradesh	5.554946e+07	162968.000000
3	Kerala	3.340606e+07	161606.333333