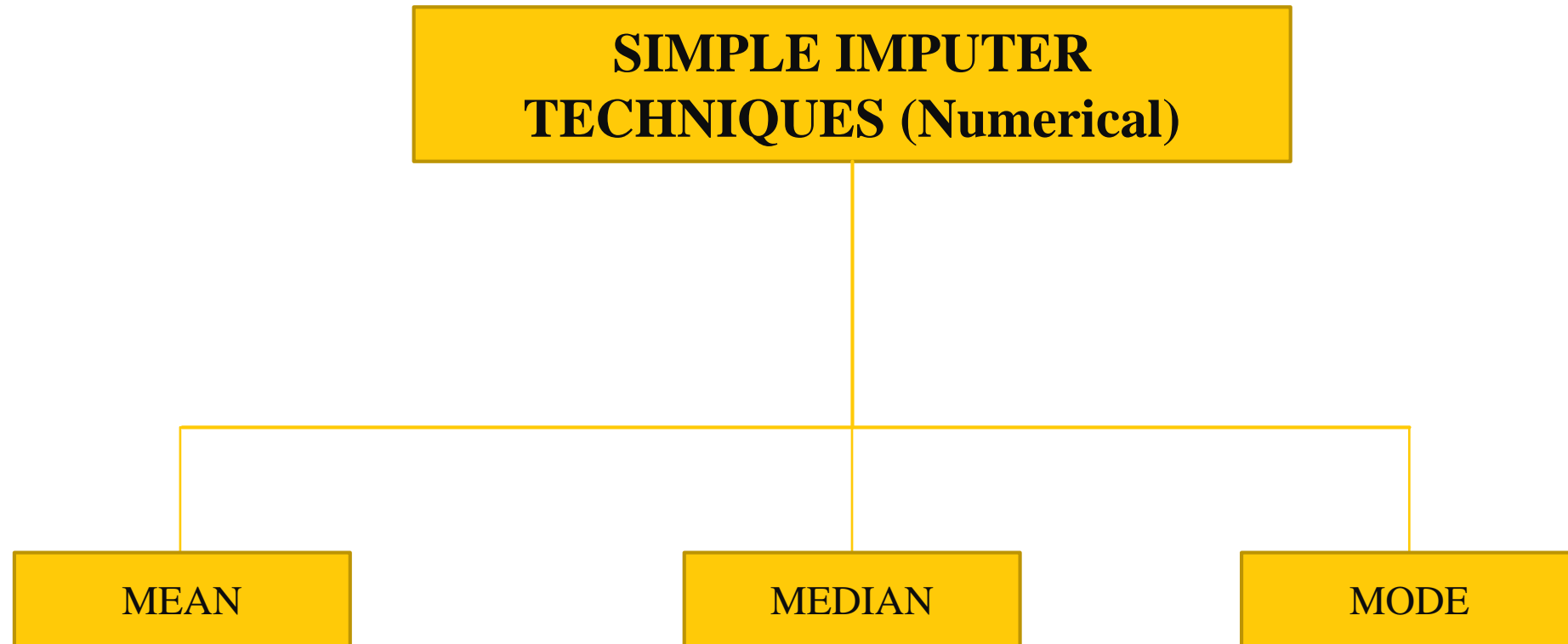




SIMPLE IMPUTER TECHNIQUE

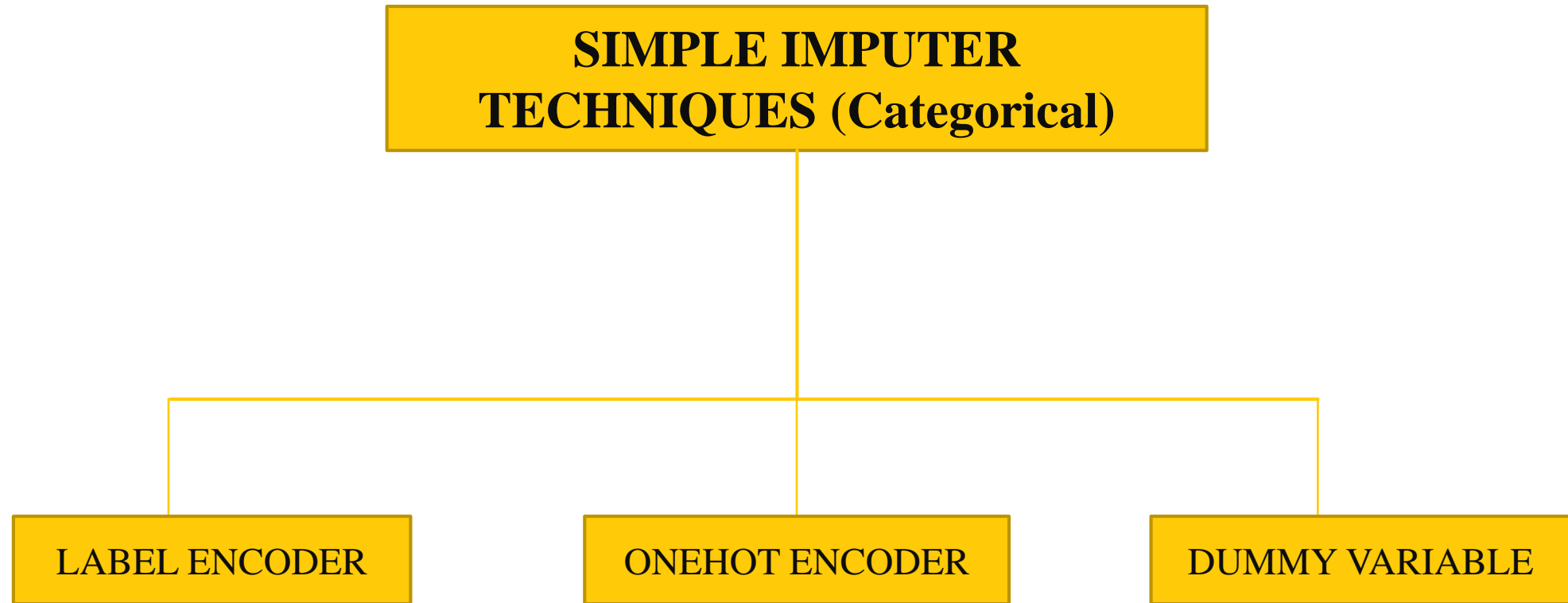
NAME : G.AKHILA
MISSING VALUE TREATMENT

SIMPLE IMPUTER TECHNIQUES (Numerical Data)



G.AKHILA
SIMPLE IMPUTER
EDA

SIMPLE IMPUTER TECHNIQUES (Numerical Data)



G.AKHILA
SIMPLE IMPUTER
EDA

SIMPLE IMPUTER

Q. Why do we use simple imputer ?

A. When we have missing values in either categorical data or numerical data we will use simple imputer to impute missing values in the dataset

DATASET FOR PRACTICE

	A	B	C	D
1	Country	Age	Salary	Purchased
2	France	44	72000	No
3	Spain	27	48000	Yes
4	Germany	30	54000	No
5	Spain	38	61000	No
6	Germany	40	←	Yes
7	France	45	58000	Yes
8	Spain	←	52000	No
9	France	48	79000	Yes
10	Germany	50	83000	No
11	France	37	67000	Yes

Missing

Missing

G.AKHILA
SIMPLE IMPUTER
EDA

PACKAGES WE NEED

- numpy
- pandas
- sklearn (model_selection, preprocessing)

G.AKHILA
SIMPLE INPUTER
EDA

CODE FOR IMPUTING MISSING VALUES USING MEAN TECHNIQUE

```
Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help

...Machine Learning\Simple Imputer (Mean,Median & Mode)\Missing Value Treatment using MEAN Strategy.py

Missing Value Treatment using MEAN Strategy.py* X

1 # Importing the libraries
2 import numpy as np
3 import pandas as pd
4 import matplotlib.pyplot as plt
5
6 # Reading the dataset & building dependent & independent variables
7 dataset = pd.read_csv('C:\Users\Akshita\Desktop\Machine Learning\Simple Imputer (Mean,Median & Mode)\dataset.csv')
8 X = dataset.iloc[:, :-1].values
9 y = dataset.iloc[:, -1].values
10
11 # Missing value treatment using whole dataset (for training) - Mean
12 from sklearn.impute import SimpleImputer
13 imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
14
15 # Fit & transform dataset - Fill the missing values of numerical data
16 imputer = imputer.fit(X[:, 1:3])
17 X[:, 1:3] = imputer.transform(X[:, 1:3])
18
19 # Reading completed data using (for testing) - Mean
20 from sklearn.preprocessing import LabelEncoder
21 labelencoder_x = LabelEncoder()
22 labelencoder_y = LabelEncoder()
23 X[:, 0] = labelencoder_x.fit_transform(X[:, 0])
24
25 labelencoder_y = LabelEncoder()
26 y = labelencoder_y.fit_transform(y)
27
28 # Splitting Training & Testing Data
29 from sklearn.model_selection import train_test_split
30
31 # Splitting the data into sets
32 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
33
34 # Feature Scaling
35 from sklearn.preprocessing import StandardScaler
36 sc_x = StandardScaler()
37 X_train = sc_x.fit_transform(X_train)
38 X_test = sc_x.transform(X_test)
```

MEAN TECHNIQUE CODE

G.AKHILA
SIMPLE IMPUTER
EDA

DATASET (BEFORE & AFTER IMPUTING MEAN TECHNIQUE & LABEL ENCODER)

dataset - DataFrame

Index	Country	Age	Salary	Purchased
0	France	44	72000	No
1	Spain	27	48000	Yes
2	Germany	30	54000	No
3	Spain	38	61000	No
4	Germany	40	nan	Yes
5	France	45	58000	Yes
6	Spain	nan	52000	No
7	France	48	79000	Yes
8	Germany	50	83000	No
9	France	37	67000	Yes

BEFORE

X - NumPy object array (read only)

	0	1	2
0	0	44.0	72000.0
1	2	27.0	48000.0
2	1	30.0	54000.0
3	2	38.0	61000.0
4	1	40.0	63777.777777...
5	0	45.0	58000.0
6	2	39.8888888888...	52000.0
7	0	48.0	79000.0
8	1	50.0	83000.0
9	0	37.0	67000.0

AFTER

**MEAN
TECHNIQUE**

G.AKHILA
SIMPLE IMPUTER
EDA

CODE FOR IMPUTING MISSING VALUES USING MEDIAN TECHNIQUE

```
Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help

...chine Learning(Simple Imputer (Mean,Median & Mode))\Missing Value Treatment using MEDIAN Strategy.py

Missing Value Treatment using MEDIAN Strategy.py X

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 dataset = pd.read_csv("../datasets/IMSLI/datasets/Machine Learning/Dataset (Mean,Median & Mode)/Mean.csv")
6 X = dataset.iloc[:,1:3].values
7 y = dataset.iloc[:,3].values
8
9 # Missing value treatment using simple imputer (for strategy = mean)
10
11 from sklearn.impute import SimpleImputer
12 imputer = SimpleImputer(missing_values=np.nan, strategy='median')
13
14 # Fit & transform method - Filled Missing values of numerical data
15
16 imputer = imputer.fit(X[:,1:3])
17 X[:,1:3] = imputer.transform(X[:,1:3])
18
19 # Encoding categorical data using Label Encoder Technique
20
21 from sklearn.preprocessing import LabelEncoder
22
23 labelencoder_x = LabelEncoder()
24 labelencoder_x.fit_transform(X[:,0])
25 X[:,0] = labelencoder_x.fit_transform(X[:,0])
26
27 labelencoder_y = LabelEncoder()
28 y = labelencoder_y.fit_transform(y)
29
30 # Splitting training & testing data
31 from sklearn.model_selection import train_test_split
32
33 # testing the model with 80%
34 X_train,X_test,y_train,y_test = train_test_split(X,y, test_size=0.2, random_state=0)
35
36 # Feature Scaling
37
38 from sklearn.preprocessing import StandardScaler
39
40 sc_x = StandardScaler()
41 X_train = sc_x.fit_transform(X_train)
42 X_test = sc_x.transform(X_test)
```

MEDIAN TECHNIQUE CODE

G.AKHILA
SIMPLE IMPUTER
EDA

DATASET (BEFORE & AFTER IMPUTING MEAN TECHNIQUE & LABEL ENCODER)

dataset - DataFrame

Index	Country	Age	Salary	Purchased
0	France	44	72000	No
1	Spain	27	48000	Yes
2	Germany	30	54000	No
3	Spain	38	61000	No
4	Germany	40	nan	Yes
5	France	45	58000	Yes
6	Spain	nan	52000	No
7	France	48	79000	Yes
8	Germany	50	83000	No
9	France	37	67000	Yes

BEFORE

X - NumPy object array (read only)

	0	1	2
0	0	44.0	72000.0
1	2	27.0	48000.0
2	1	30.0	54000.0
3	2	38.0	61000.0
4	1	40.0	61000.0
5	0	45.0	58000.0
6	2	40.0	52000.0
7	0	48.0	79000.0
8	1	50.0	83000.0
9	0	37.0	67000.0

AFTER

**MEDIAN
TECHNIQUE**

G.AKHILA
SIMPLE IMPUTER
EDA

CODE FOR IMPUTING MISSING VALUES USING MODE TECHNIQUE

```
Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help

...Machine Learning\Simple Imputer (Mean,Median & Mode)\Missing Value Treatment using MODE

Missing Value Treatment using MODE Strategy.py* X

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 dataset = pd.read_csv("C:\Users\AKHILA\Desktop\dataset\Machine Learning\Simple Imputer (Mean,Median & Mode)\Mean.csv")
6 X = dataset.iloc[:,1:4].values
7 y = dataset.iloc[:,4].values
8
9 # Missing value treatment using simple imputer (fit strategy = Mean)
10
11 from sklearn.impute import SimpleImputer
12 imputer = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
13
14 #
15
16 # Fit & Transform method - Filled Missing Values of Numerical Data
17
18 imputer = imputer.fit(X[:,1:3])
19 X[:,1:3] = imputer.transform(X[:,1:3])
20
21 #
22
23 # Encoding Categorical Data using LabelEncoder, ONE-HEAT Technique
24
25 from sklearn.preprocessing import LabelEncoder
26 labelencoder_X = LabelEncoder()
27 labelencoder_X.fit_transform(X[:,0])
28 X[:,0] = labelencoder_X.fit_transform(X[:,0])
29
30 #
31
32 labelencoder_y = LabelEncoder()
33 y = labelencoder_y.fit_transform(y)
34
35 #
36
37 # Splitting Training & Testing Phase
38
39 from sklearn.model_selection import train_test_split
40
41 # testing the model with all
42 X_train,X_test,y_train,y_test = train_test_split(X,y, test_size=0.2, random_state=0)
43
44 #
45
46 # Feature Scaling
47
48 from sklearn.preprocessing import StandardScaler
49 sc_X = StandardScaler()
50 X_train = sc_X.fit_transform(X_train)
51 X_test = sc_X.transform(X_test)
```

MODE TECHNIQUE CODE

G.AKHILA
SIMPLE IMPUTER
EDA

DATASET (BEFORE & AFTER IMPUTING MEAN TECHNIQUE & LABEL ENCODER)

dataset - DataFrame

Index	Country	Age	Salary	Purchased
0	France	44	72000	No
1	Spain	27	48000	Yes
2	Germany	30	54000	No
3	Spain	38	61000	No
4	Germany	40	nan	Yes
5	France	45	58000	Yes
6	Spain	nan	52000	No
7	France	48	79000	Yes
8	Germany	50	83000	No
9	France	37	67000	Yes

BEFORE

X - NumPy object array (read only)

	0	1	2
0	0	44.0	72000.0
1	2	27.0	48000.0
2	1	30.0	54000.0
3	2	38.0	61000.0
4	1	40.0	48000.0
5	0	45.0	58000.0
6	2	27.0	52000.0
7	0	48.0	79000.0
8	1	50.0	83000.0
9	0	37.0	67000.0

AFTER

**MODE
TECHNIQUE**

G.AKHILA
SIMPLE IMPUTER
EDA