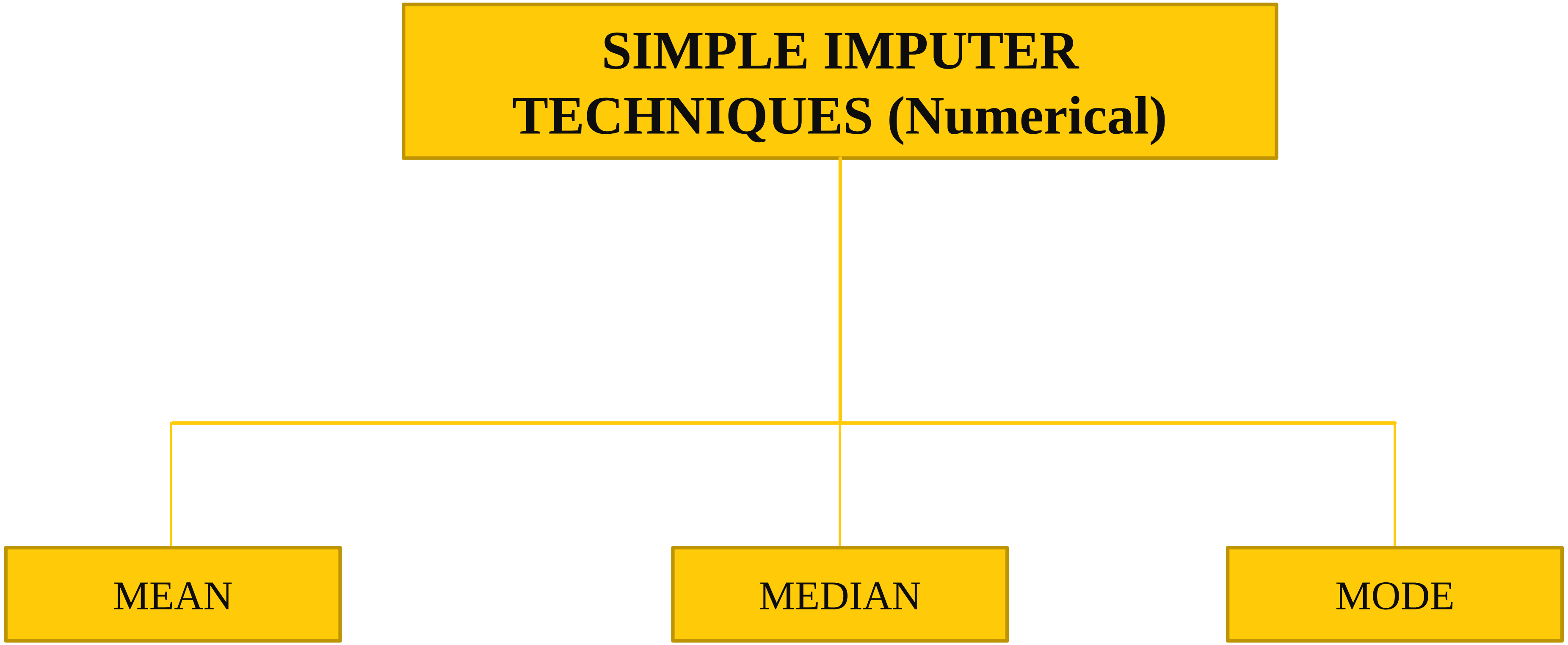
SIMPLE IMPUTER TECHNIQUE

NAME : G.AKHILA
MISSING VALUE TREATMENT

SIMPLE IMPUTER TECHNIQUES (Numerical Data)

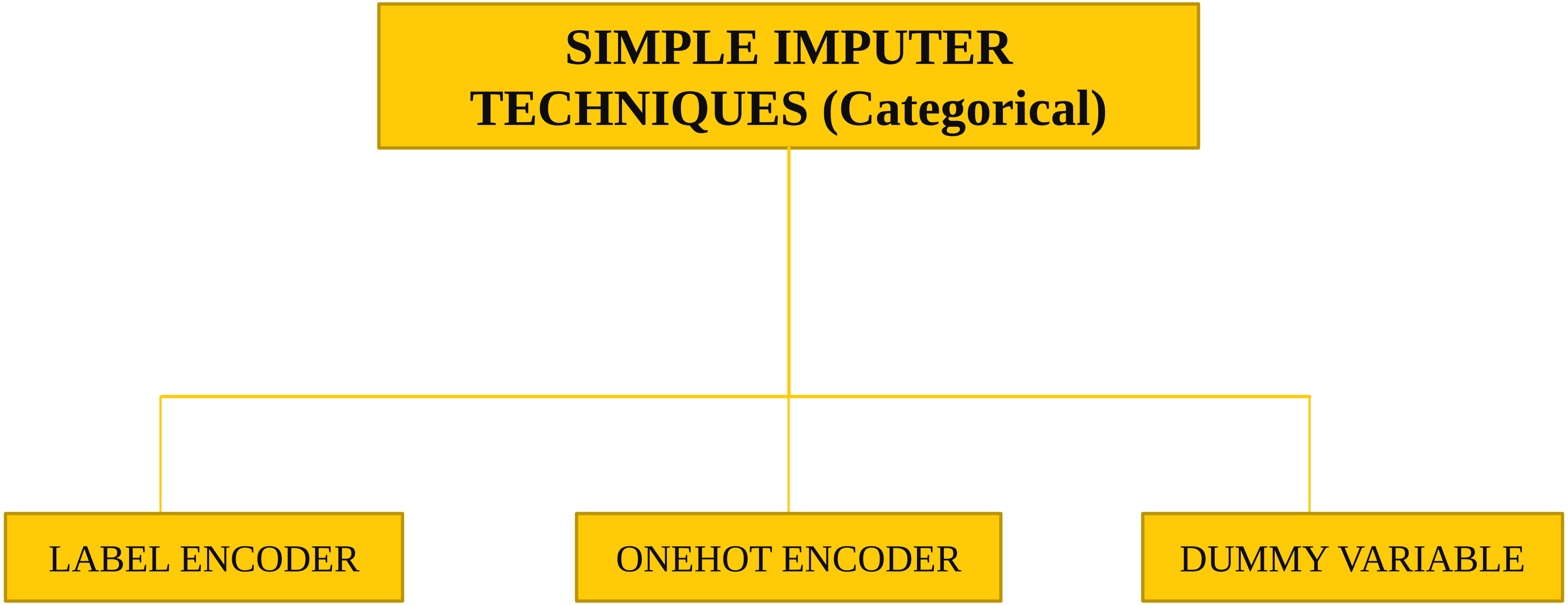

SIMPLE IMPUTER TECHNIQUES (Numerical)

MEAN

MEDIAN

MODE

SIMPLE IMPUTER TECHNIQUES (Numerical Data)



SIMPLE IMPUTER TECHNIQUES (Categorical)

LABEL ENCODER

ONEHOT ENCODER

DUMMY VARIABLE

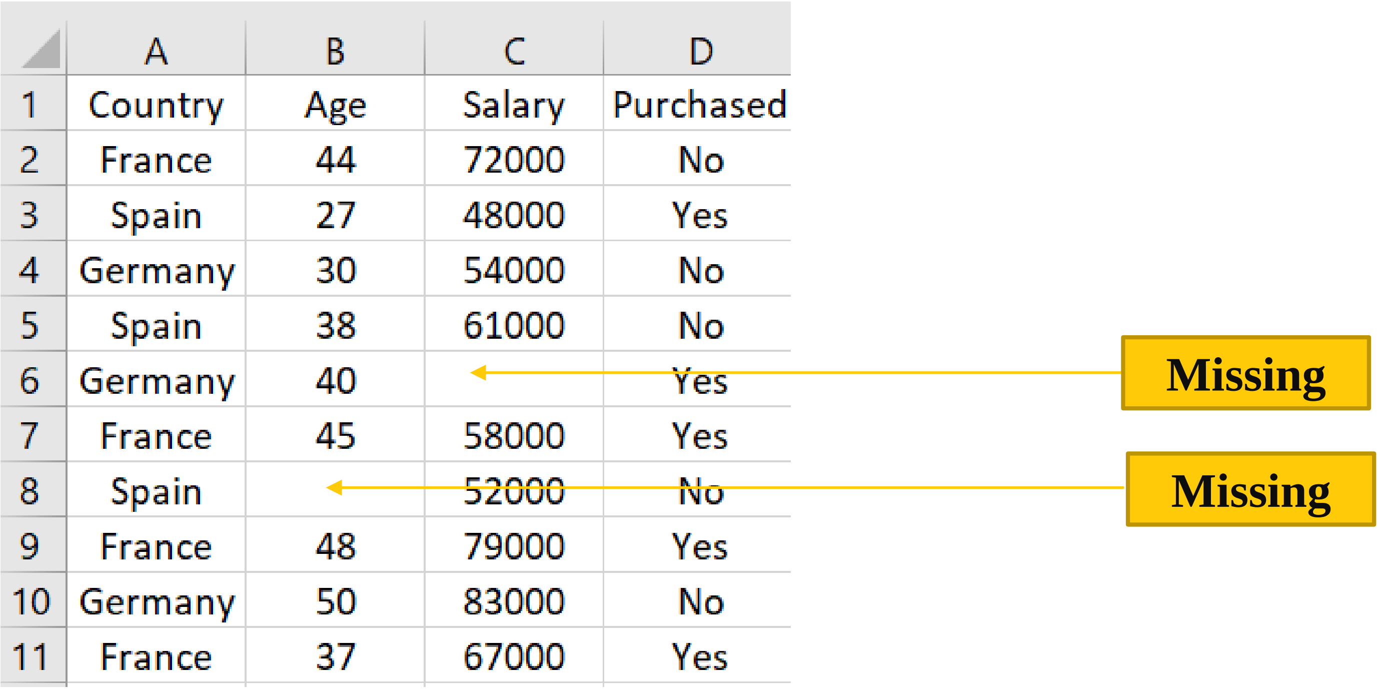
SIMPLE IMPUTER

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

G.AKHILA
SIMPLE IMPUTER
EDA

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

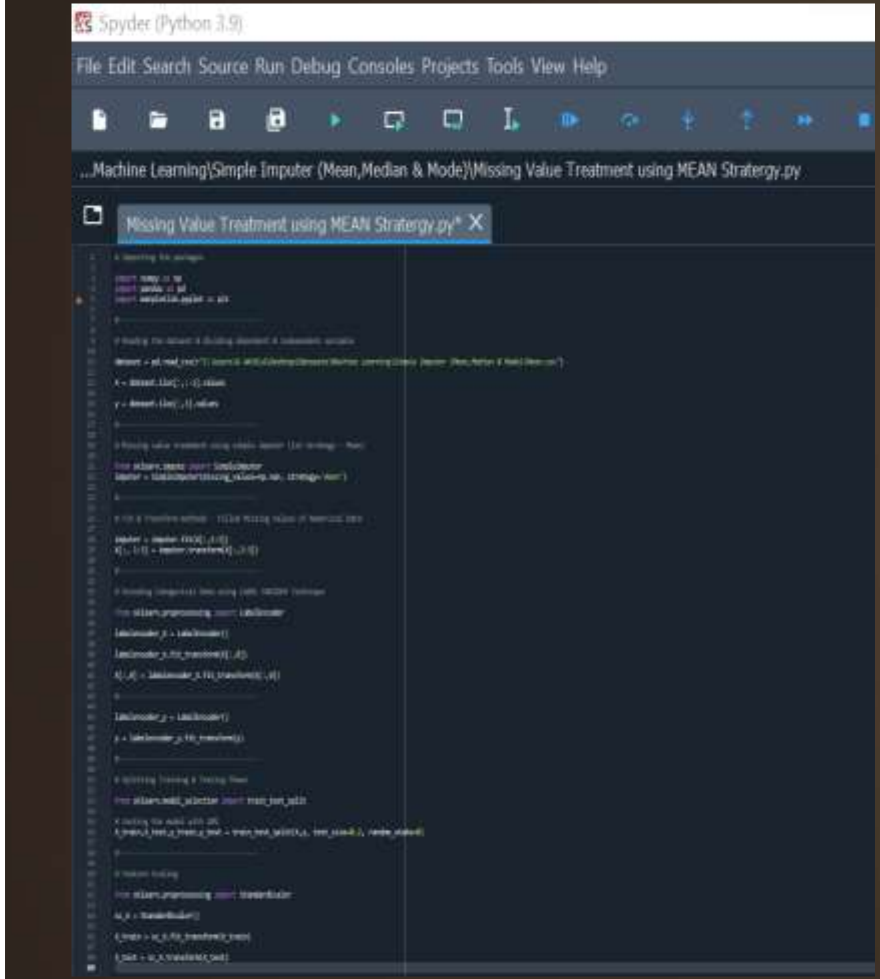
G.AKHILA
SIMPLE IMPUTER
EDA

PACKAGES WE NEED

- numpy
- pandas
- sklearn (model_selection, preprocessing)

G.AKHILA
SIMPLE IMPUTER
EDA

CODE FOR IMPUTING MISSING VALUES USING MEAN TECHNIQUE

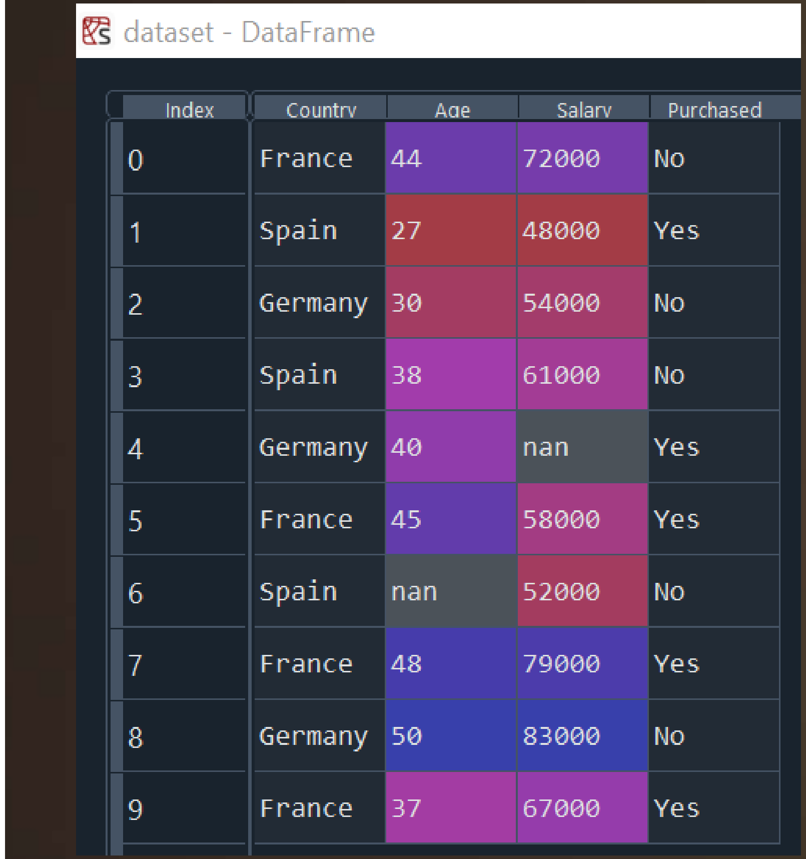

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
...Machine Learning\Simple Imputer (Mean,Median & Mode)\Missing Value Treatment using MEAN Stratergy.py
Missing Value Treatment using MEAN Stratergy.py*

```
# Importing the packages

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

#-----------------------------

# Reading the dataset & dividing dependent & independent variable

dataset = pd.read_csv(r"[illegible]")

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:,3].values

#-----------------------------

# Missing value treatment using simple imputer (i.e strategy = Mean)

from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')

#-----------------------------

# Fit & Transform method - filled missing values of numerical data

imputer = imputer.fit(X[:,1:3])
X[:, 1:3] = imputer.transform(X[:,1:3])

#-----------------------------

# Encoding categorical data using LABEL ENCODER technique

from sklearn.preprocessing import LabelEncoder

labelencoder_X = LabelEncoder()

labelencoder_X.fit_transform(X[:,0])

X[:,0] = labelencoder_X.fit_transform(X[:,0])

#-----------------------------

labelencoder_y = LabelEncoder()

y = labelencoder_y.fit_transform(y)

#-----------------------------

# Splitting Training & Testing Rows

from sklearn.model_selection import train_test_split

# testing the model with 20%
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size=0.2, random_state=0)

#-----------------------------

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc_X = StandardScaler()

X_train = sc_X.fit_transform(X_train)

X_test = sc_X.transform(X_test)
```

MEAN
TECHNIQUE CODE

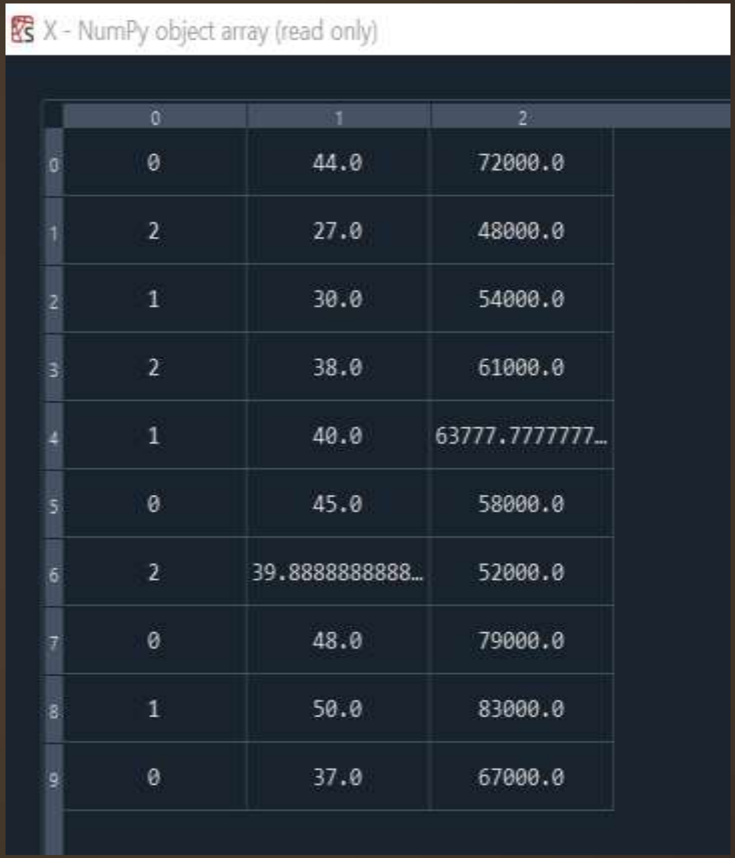
G.AKHILA
SIMPLE IMPUTER
EDA

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



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



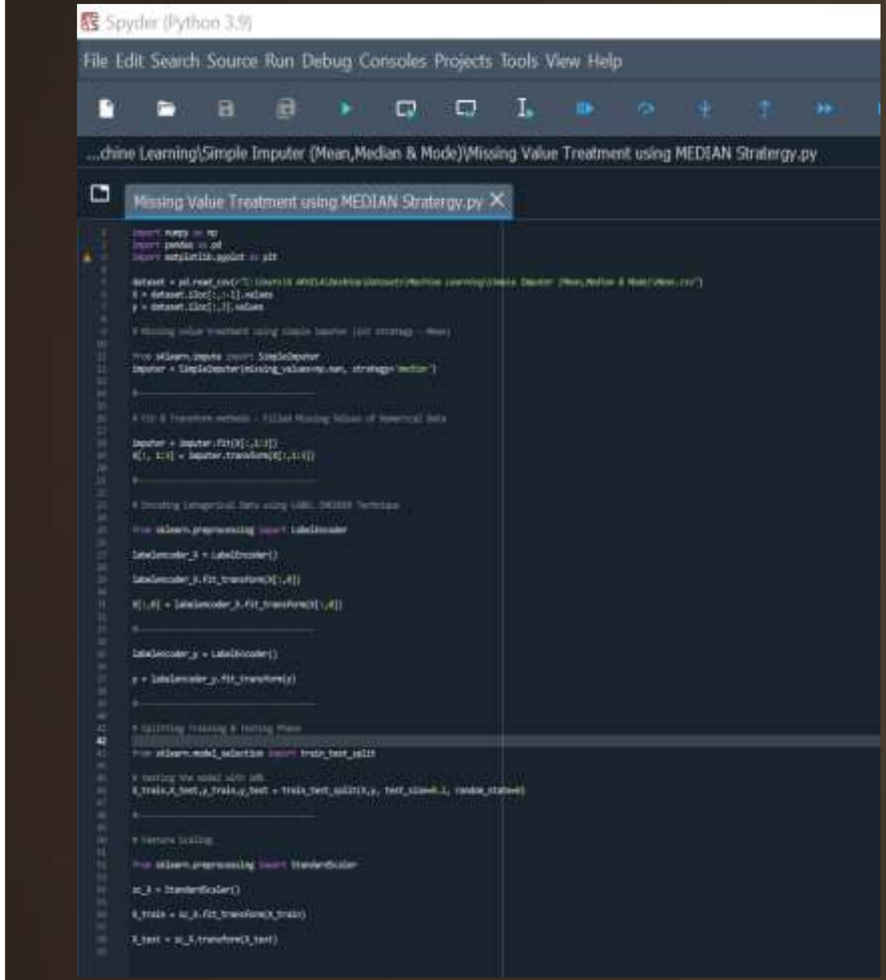
	0	1	2
0	0	44.0	72000.0
1	2	27.0	48000.0
2	1	30.0	54000.0
3	2	38.0	61000.0
4	1	40.0	63777.7777777…
5	0	45.0	58000.0
6	2	39.8888888888…	52000.0
7	0	48.0	79000.0
8	1	50.0	83000.0
9	0	37.0	67000.0

AFTER

MEAN TECHNIQUE

G.AKHILA
SIMPLE IMPUTER
EDA

CODE FOR IMPUTING MISSING VALUES USING MEDIAN TECHNIQUE

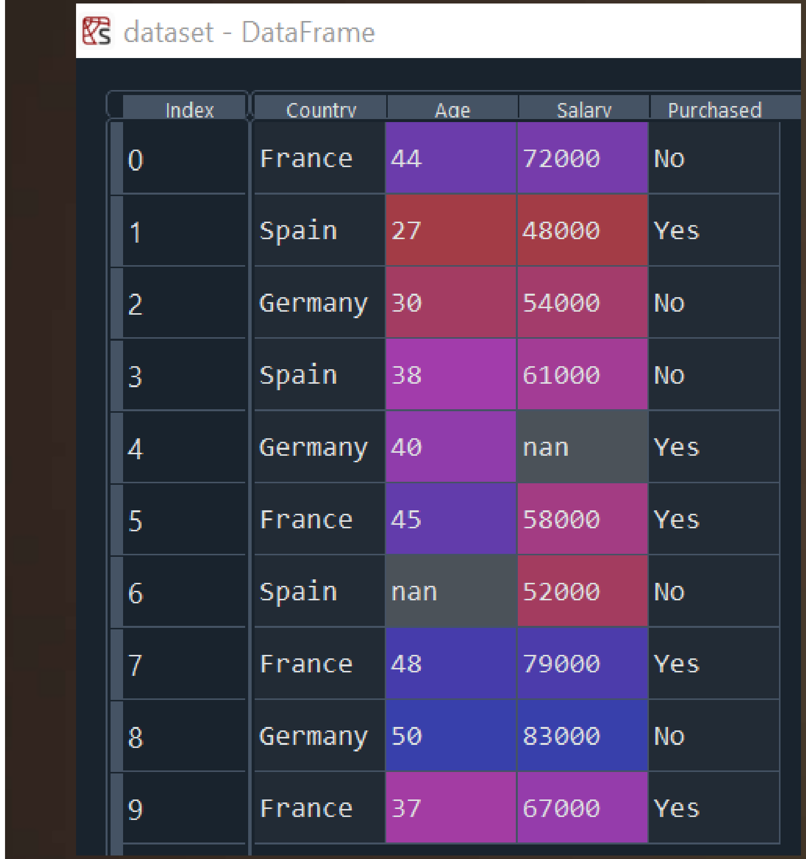

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
...chine Learning\Simple Imputer (Mean,Median & Mode)\Missing Value Treatment using MEDIAN Stratergy.py
Missing Value Treatment using MEDIAN Stratergy.py

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

dataset = pd.read_csv('C:\Users\[illegible]\Desktop\Datasets\Machine Learning\Simple Imputer (Mean,Median & Mode)\Mean.csv')
X = dataset.iloc[:,:-1].values
y = dataset.iloc[:,3].values

# Missing value treatment using simple imputer (dt strategy - Mean)

from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='median')

#------------------------------------------

# Fit & Transform methods - Filled Missing Values of Numerical Data

imputer = imputer.fit(X[:,1:3])
X[:, 1:3] = imputer.transform(X[:,1:3])

#------------------------------------------

# Encoding Categorical Data using LABEL ENCODER Technique

from sklearn.preprocessing import LabelEncoder

labelencoder_X = LabelEncoder()

labelencoder_X.fit_transform(X[:,0])

X[:,0] = labelencoder_X.fit_transform(X[:,0])

#------------------------------------------

labelencoder_y = LabelEncoder()

y = labelencoder_y.fit_transform(y)

#------------------------------------------

# Splitting Training & Testing Phase

from sklearn.model_selection import train_test_split

# testing the model with 10%
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size=0.1, random_state=0)

#------------------------------------------

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc_X = StandardScaler()

X_train = sc_X.fit_transform(X_train)

X_test = sc_X.transform(X_test)
```

MEDIAN TECHNIQUE CODE

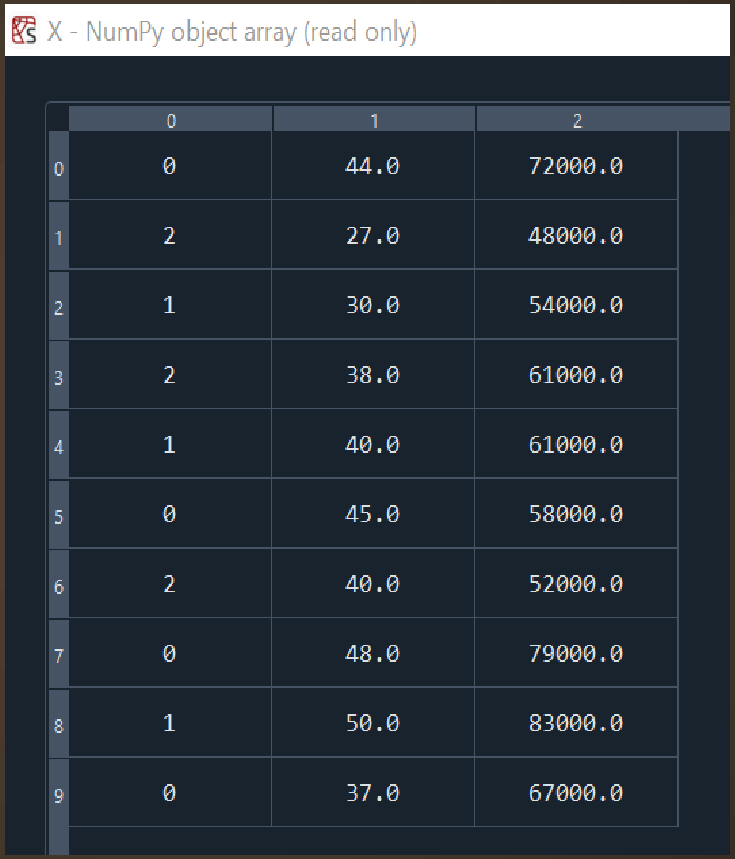
G.AKHILA
SIMPLE IMPUTER
EDA

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



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



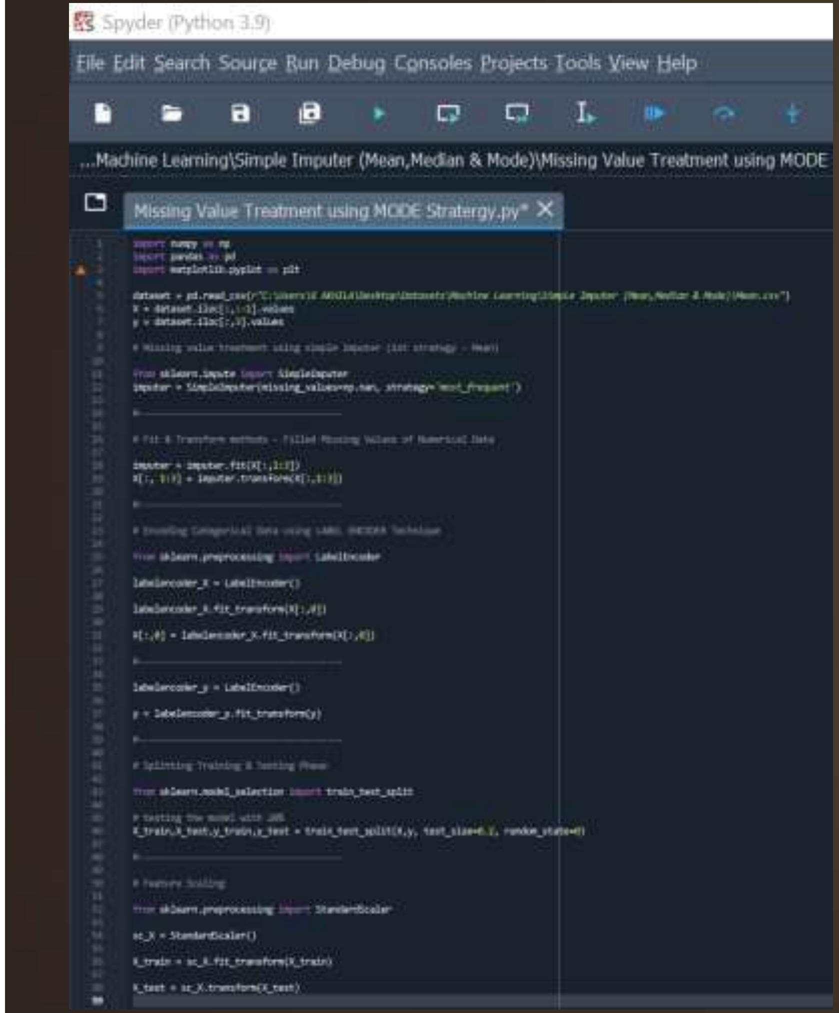
	0	1	2
0	0	44.0	72000.0
1	2	27.0	48000.0
2	1	30.0	54000.0
3	2	38.0	61000.0
4	1	40.0	61000.0
5	0	45.0	58000.0
6	2	40.0	52000.0
7	0	48.0	79000.0
8	1	50.0	83000.0
9	0	37.0	67000.0

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

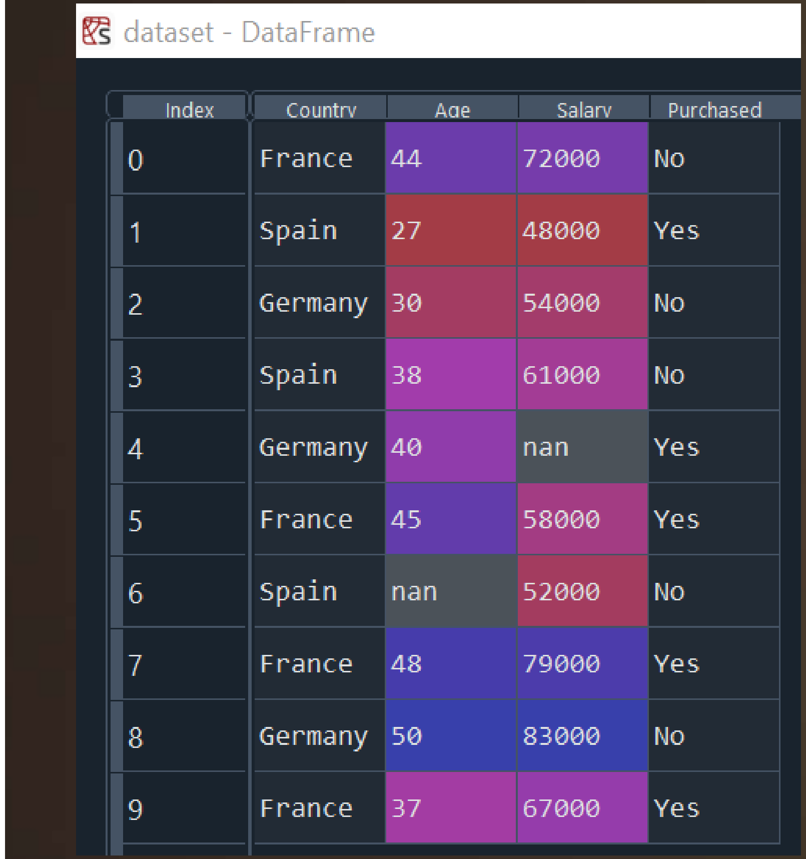
Missing Value Treatment using MODE Stratergy.py*

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

dataset = pd.read_csv(r"C:\Users\G AKHILA\Desktop\Datasets\Machine Learning\Simple Imputer (Mean,Median & Mode)\Mean.csv")
X = dataset.iloc[:,:-1].values
y = dataset.iloc[:,3].values

# Missing value treatment using simple imputer (1st strategy - Mean)

from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='most_frequent')

#%%

# Fit & Transform methods - Filled Missing Values of Numerical Data

imputer = imputer.fit(X[:,1:3])
X[:, 1:3] = imputer.transform(X[:,1:3])

#%%

# Encoding Categorical Data using LABEL ENCODER Technique

from sklearn.preprocessing import LabelEncoder

labelencoder_X = LabelEncoder()

labelencoder_X.fit_transform(X[:,0])

X[:,0] = labelencoder_X.fit_transform(X[:,0])

#%%

labelencoder_y = LabelEncoder()

y = labelencoder_y.fit_transform(y)

#%%

# Splitting Training & Testing Phase

from sklearn.model_selection import train_test_split

# testing the model with 20%
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size=0.2, random_state=0)

#%%

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc_X = StandardScaler()

X_train = sc_X.fit_transform(X_train)

X_test = sc_X.transform(X_test)
```

MODE
TECHNIQUE CODE

G.AKHILA
SIMPLE IMPUTER
EDA

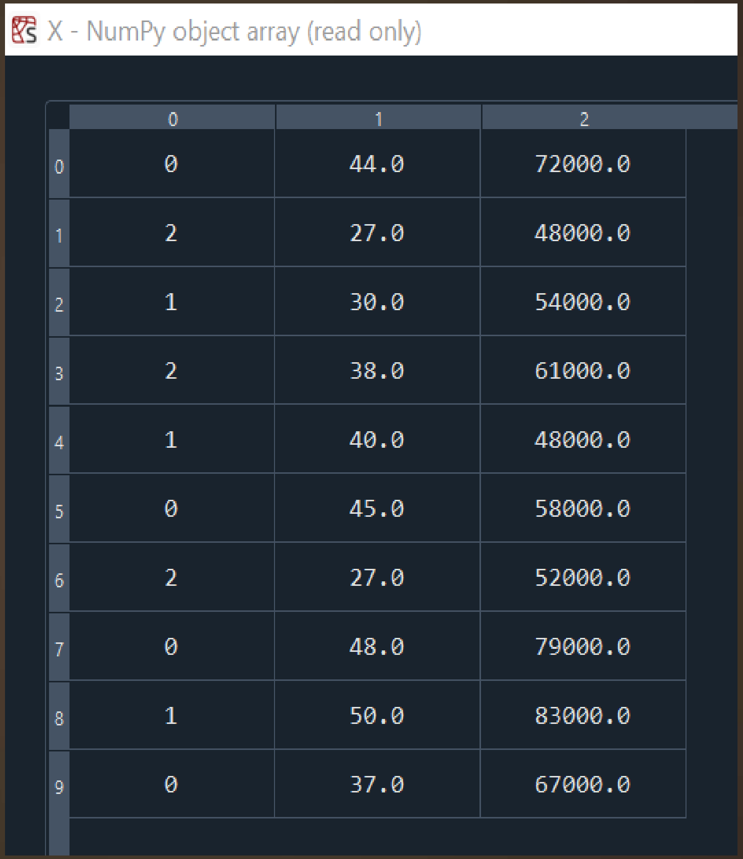
DATASET (BEFORE & AFTER IMPUTING MEAN TECHNIQUE & LABEL ENCODER)



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