Practical Lab Assignment 3

Pre-processing of missing values (using mean value of each attribute class):

Problem Statement: Replace the missing values for given automobile dataset “imports-85.data” with mean value of each attribute class. (Consider no. of doors as the class attribute - 6th attribute)

Create a Menu Driven Python program.

Dataset:
https://github.com/nyuvis/datasets/blob/master/auto/imports-85.data

Dataset Information :
https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.names

Implementation:

```
#import the libraries
import pandas as pd
import numpy as np
from google.colab import files

#import the dataset
ds = pd.read_csv("https://raw.githubusercontent.com/nyuvis/datasets/master/auto/i
mports-85.data")

#replace the NaN value for "?" value
ds.replace("?",np.nan,inplace=True)

# convert the object class into float object
ds['normalized-losses'] = ds['normalized-losses'].astype(float)
ds['bore'] = ds['bore'].astype(float)
ds['stroke'] = ds['stroke'].astype(float)
ds['horsepower'] = ds['horsepower'].astype(float)
ds['peak-rpm'] = ds['peak-rpm'].astype(float)
ds['price'] = ds['price'].astype(float)

#creating the menu driven program
print("=============MENU=============")
print("1. Missing Values using MEAN")
print("2. Missing Values using MEDIAN")
print("3. Missing Values using MODE")
option = int(input("Enter Choice :- "))



if option == 1:

  #taking the individual mean by taking num-of-doors as class
  mean_t_nod = ds[ds['num-of-doors'] == 'two']['normalized-losses'].mean()
  mean_f_nod = ds[ds['num-of-doors'] == 'four']['normalized-losses'].mean()

  mean_t_bore = ds[ds['num-of-doors'] == 'two']['bore'].mean()
  mean_f_bore = ds[ds['num-of-doors'] == 'four']['bore'].mean()

  mean_t_stroke = ds[ds['num-of-doors'] == 'two']['stroke'].mean()
  mean_f_stroke = ds[ds['num-of-doors'] == 'four']['stroke'].mean()

  mean_t_horsepower = ds[ds['num-of-doors'] == 'two']['horsepower'].mean()
```

```
  mean_f_horsepower = ds[ds['num-of-doors'] == 'four']['horsepower'].mean()

  mean_t_peak_rpm = ds[ds['num-of-doors'] == 'two']['peak-rpm'].mean()
  mean_f_peak_rpm = ds[ds['num-of-doors'] == 'four']['peak-rpm'].mean()

  mean_t_price = ds[ds['num-of-doors'] == 'two']['price'].mean()
  mean_f_price = ds[ds['num-of-doors'] == 'four']['price'].mean()


  #filling all the missing value by mean value of each attribute class
  ds['normalized-losses'][ds['num-of-doors'] == 'two'] = ds['normalized-
losses'][ds['num-of-doors'] == 'two'].fillna(mean_t_nod)
  ds['normalized-losses'][ds['num-of-doors'] == 'four'] = ds['normalized-
losses'][ds['num-of-doors'] == 'four'].fillna(mean_f_nod)

  ds['bore'][ds['num-of-doors'] == 'two'] = ds['bore'][ds['num-of-
doors'] == 'two'].fillna(mean_t_bore)
  ds['bore'][ds['num-of-doors'] == 'four'] = ds['bore'][ds['num-of-
doors'] == 'four'].fillna(mean_f_bore)

  ds['stroke'][ds['num-of-doors'] == 'two'] = ds['stroke'][ds['num-of-
doors'] == 'two'].fillna(mean_t_stroke)
  ds['stroke'][ds['num-of-doors'] == 'four'] = ds['stroke'][ds['num-of-
doors'] == 'four'].fillna(mean_f_stroke)

  ds['horsepower'][ds['num-of-doors'] == 'two'] = ds['horsepower'][ds['num-of-
doors'] == 'two'].fillna(mean_t_horsepower)
  ds['horsepower'][ds['num-of-doors'] == 'four'] = ds['horsepower'][ds['num-of-
doors'] == 'four'].fillna(mean_f_horsepower)

  ds['peak-rpm'][ds['num-of-doors'] == 'two'] = ds['peak-rpm'][ds['num-of-
doors'] == 'two'].fillna(mean_t_peak_rpm)
  ds['peak-rpm'][ds['num-of-doors'] == 'four'] = ds['peak-rpm'][ds['num-of-
doors'] == 'four'].fillna(mean_f_peak_rpm)

  ds['price'][ds['num-of-doors'] == 'two'] = ds['price'][ds['num-of-
doors'] == 'two'].fillna(mean_t_price)
  ds['price'][ds['num-of-doors'] == 'four'] = ds['price'][ds['num-of-
doors'] == 'four'].fillna(mean_f_price)

  #this converts the dataset into CSV files and then download it.
  ds.to_csv('mean.csv')
  files.download('mean.csv')
```

```
elif option == 2:

  #taking the individual median by taking num-of-doors as class
  median_t_nod = ds[ds['num-of-doors'] == 'two']['normalized-losses'].median()
  median_f_nod = ds[ds['num-of-doors'] == 'four']['normalized-losses'].median()

  median_t_bore = ds[ds['num-of-doors'] == 'two']['bore'].median()
  median_f_bore = ds[ds['num-of-doors'] == 'four']['bore'].median()

  median_t_stroke = ds[ds['num-of-doors'] == 'two']['stroke'].median()
  median_f_stroke = ds[ds['num-of-doors'] == 'four']['stroke'].median()

  median_t_horsepower = ds[ds['num-of-doors'] == 'two']['horsepower'].median()
  median_f_horsepower = ds[ds['num-of-doors'] == 'four']['horsepower'].median()

  median_t_peak_rpm = ds[ds['num-of-doors'] == 'two']['peak-rpm'].median()
  median_f_peak_rpm = ds[ds['num-of-doors'] == 'four']['peak-rpm'].median()

  median_t_price = ds[ds['num-of-doors'] == 'two']['price'].median()
  median_f_price = ds[ds['num-of-doors'] == 'four']['price'].median()

  #filling all the missing value by median value of each attribute class
  ds['normalized-losses'][ds['num-of-doors'] == 'two'] = ds['normalized-
losses'][ds['num-of-doors'] == 'two'].fillna(median_t_nod)
  ds['normalized-losses'][ds['num-of-doors'] == 'four'] = ds['normalized-
losses'][ds['num-of-doors'] == 'four'].fillna(median_f_nod)

  ds['bore'][ds['num-of-doors'] == 'two'] = ds['bore'][ds['num-of-
doors'] == 'two'].fillna(median_t_bore)
  ds['bore'][ds['num-of-doors'] == 'four'] = ds['bore'][ds['num-of-
doors'] == 'four'].fillna(median_f_bore)

  ds['stroke'][ds['num-of-doors'] == 'two'] = ds['stroke'][ds['num-of-
doors'] == 'two'].fillna(median_t_stroke)
  ds['stroke'][ds['num-of-doors'] == 'four'] = ds['stroke'][ds['num-of-
doors'] == 'four'].fillna(median_f_stroke)

  ds['horsepower'][ds['num-of-doors'] == 'two'] = ds['horsepower'][ds['num-of-
doors'] == 'two'].fillna(median_t_horsepower)
  ds['horsepower'][ds['num-of-doors'] == 'four'] = ds['horsepower'][ds['num-of-
doors'] == 'four'].fillna(median_f_horsepower)
```

```
  ds['peak-rpm'][ds['num-of-doors'] == 'two'] = ds['peak-rpm'][ds['num-of-
doors'] == 'two'].fillna(median_t_peak_rpm)
  ds['peak-rpm'][ds['num-of-doors'] == 'four'] = ds['peak-rpm'][ds['num-of-
doors'] == 'four'].fillna(median_f_peak_rpm)

  ds['price'][ds['num-of-doors'] == 'two'] = ds['price'][ds['num-of-
doors'] == 'two'].fillna(median_t_price)
  ds['price'][ds['num-of-doors'] == 'four'] = ds['price'][ds['num-of-
doors'] == 'four'].fillna(median_f_price)

  #this converts the dataset into CSV files and then download it.
  ds.to_csv('median.csv')
  files.download('median.csv')



elif option == 3:

  #taking the individual mode by taking num-of-doors as class
  mode_t_nod = ds[ds['num-of-doors'] == 'two']['normalized-losses'].mode()[0]
  mode_f_nod = ds[ds['num-of-doors'] == 'four']['normalized-losses'].mode()[0]

  mode_t_bore = ds[ds['num-of-doors'] == 'two']['bore'].mode()[0]
  mode_f_bore = ds[ds['num-of-doors'] == 'four']['bore'].mode()[0]

  mode_t_stroke = ds[ds['num-of-doors'] == 'two']['stroke'].mode()[0]
  mode_f_stroke = ds[ds['num-of-doors'] == 'four']['stroke'].mode()[0]

  mode_t_horsepower = ds[ds['num-of-doors'] == 'two']['horsepower'].mode()[0]
  mode_f_horsepower = ds[ds['num-of-doors'] == 'four']['horsepower'].mode()[0]

  mode_t_peak_rpm = ds[ds['num-of-doors'] == 'two']['peak-rpm'].mode()[0]
  mode_f_peak_rpm = ds[ds['num-of-doors'] == 'four']['peak-rpm'].mode()[0]

  mode_t_price = ds[ds['num-of-doors'] == 'two']['price'].mode()[0]
  mode_f_price = ds[ds['num-of-doors'] == 'four']['price'].mode()[0]

  #filling all the missing value by mode value of each attribute class
  ds['normalized-losses'][ds['num-of-doors'] == 'two'] = ds['normalized-
losses'][ds['num-of-doors'] == 'two'].fillna(mode_t_nod)
  ds['normalized-losses'][ds['num-of-doors'] == 'four'] = ds['normalized-
losses'][ds['num-of-doors'] == 'four'].fillna(mode_f_nod)

  ds['bore'][ds['num-of-doors'] == 'two'] = ds['bore'][ds['num-of-
doors'] == 'two'].fillna(mode_t_bore)
```

```
  ds['bore'][ds['num-of-doors'] == 'four'] = ds['bore'][ds['num-of-
doors'] == 'four'].fillna(mode_f_bore)

  ds['stroke'][ds['num-of-doors'] == 'two'] = ds['stroke'][ds['num-of-
doors'] == 'two'].fillna(mode_t_stroke)
  ds['stroke'][ds['num-of-doors'] == 'four'] = ds['stroke'][ds['num-of-
doors'] == 'four'].fillna(mode_f_stroke)

  ds['horsepower'][ds['num-of-doors'] == 'two'] = ds['horsepower'][ds['num-of-
doors'] == 'two'].fillna(mode_t_horsepower)
  ds['horsepower'][ds['num-of-doors'] == 'four'] = ds['horsepower'][ds['num-of-
doors'] == 'four'].fillna(mode_f_horsepower)

  ds['peak-rpm'][ds['num-of-doors'] == 'two'] = ds['peak-rpm'][ds['num-of-
doors'] == 'two'].fillna(mode_t_peak_rpm)
  ds['peak-rpm'][ds['num-of-doors'] == 'four'] = ds['peak-rpm'][ds['num-of-
doors'] == 'four'].fillna(mode_f_peak_rpm)

  ds['price'][ds['num-of-doors'] == 'two'] = ds['price'][ds['num-of-
doors'] == 'two'].fillna(mode_t_price)
  ds['price'][ds['num-of-doors'] == 'four'] = ds['price'][ds['num-of-
doors'] == 'four'].fillna(mode_f_price)

  #this converts the dataset into CSV files and then download it.
  ds.to_csv('mode.csv')
  files.download('mode.csv')
```

Output:

```
=============MENU=============
1. Missing Values using MEAN
2. Missing Values using MEDIAN
3. Missing Values using MODE
Enter Choice :-
```

For Value 1 (Mean):

Whole Processed Dataset:
https://github.com/darshanjoshi16/DataMiningPracticals/blob/main/mean_after.csv

For Value 2 (Median):

Whole Processed Dataset:
https://github.com/darshanjoshi16/DataMiningPracticals/blob/main/median_after.csv

For Value 3 (Mode):

Whole Processed Dataset:
https://github.com/darshanjoshi16/DataMiningPracticals/blob/main/mode_after.csv

Sample Screenshots:

Mean CSV

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors
0	3	138.86764705882354	alfa-romero	gas	std	two
1	3	138.86764705882354	alfa-romero	gas	std	two
2	1	138.86764705882354	alfa-romero	gas	std	two
3	2	164.0	audi	gas	std	four
4	2	164.0	audi	gas	std	four
5	2	138.86764705882354	audi	gas	std	two
6	1	158.0	audi	gas	std	four
7	1	109.65263157894736	audi	gas	std	four
8	1	158.0	audi	gas	turbo	four
9	0	138.86764705882354	audi	gas	turbo	two
10	2	192.0	bmw	gas	std	two
11	0	192.0	bmw	gas	std	four
12	0	188.0	bmw	gas	std	two
13	0	188.0	bmw	gas	std	four

Median CSV

14	1	109.65263157894736	bmw	gas	std	four
15	0	109.65263157894736	bmw	gas	std	four
16	0	138.86764705882354	bmw	gas	std	two
17	0	109.65263157894736	bmw	gas	std	four
18	2	121.0	chevrolet	gas	std	two
19	1	98.0	chevrolet	gas	std	two
20	0	81.0	chevrolet	gas	std	four
21	1	118.0	dodge	gas	std	two
22	1	118.0	dodge	gas	std	two
23	1	118.0	dodge	gas	turbo	two
24	1	148.0	dodge	gas	std	four

Mode CSV

25	1	148.0	dodge	gas
26	1	148.0	dodge	gas
27	1	148.0	dodge	gas
28	-1	110.0	dodge	gas
29	3	145.0	dodge	gas
30	2	137.0	honda	gas
31	2	137.0	honda	gas
32	1	101.0	honda	gas
33	1	101.0	honda	gas