20bsit154-ass-1

March 30, 2023

[1]:
```
#1.         Get an overview of the dataset
#1.1         Read the .csv file
import pandas as pd
df= pd.read_csv("D:/sem6/dma/practical/Automobile_data.csv")
print(df)
```

```
    index      company   body-style  wheel-base  length engine-type  \
0       0  alfa-romero  convertible        88.6   168.8        dohc
1       1  alfa-romero  convertible        88.6   168.8        dohc
2       2  alfa-romero    hatchback        94.5   171.2        ohcv
3       3         audi        sedan        99.8   176.6         ohc
4       4         audi        sedan        99.4   176.6         ohc
..    …           …           …         …      …          …
56     81   volkswagen        sedan        97.3   171.7         ohc
57     82   volkswagen        sedan        97.3   171.7         ohc
58     86   volkswagen        sedan        97.3   171.7         ohc
59     87        volvo        sedan       104.3   188.8         ohc
60     88        volvo        wagon       104.3   188.8         ohc

   num-of-cylinders  horsepower  average-mileage    price
0              four         111               21  13495.0
1              four         111               21  16500.0
2               six         154               19  16500.0
3              four         102               24  13950.0
4              five         115               18  17450.0
..              …          …              …       …
56             four          85               27   7975.0
57             four          52               37   7995.0
58             four         100               26   9995.0
59             four         114               23  12940.0
60             four         114               23  13415.0

[61 rows x 10 columns]
```

[2]:
```
#1.2 Read .csv file and consider "na", "N/A", "Not Available" as set of missing␣
 ↪values

na1=["n.a.","Not Available","N/a","na"]
```

```
df= pd.read_csv("D:/sem6/dma/practical/Automobile_data.csv",na_values=na1)
print(df)
```

```
    index      company   body-style  wheel-base  length engine-type  \
0       0  alfa-romero  convertible        88.6   168.8        dohc
1       1  alfa-romero  convertible        88.6   168.8        dohc
2       2  alfa-romero    hatchback        94.5   171.2        ohcv
3       3         audi        sedan        99.8   176.6         ohc
4       4         audi        sedan        99.4   176.6         ohc
..    ...          ...          ...         ...     ...         ...
56     81   volkswagen        sedan        97.3   171.7         ohc
57     82   volkswagen        sedan        97.3   171.7         ohc
58     86   volkswagen        sedan        97.3   171.7         ohc
59     87        volvo        sedan       104.3   188.8         ohc
60     88        volvo        wagon       104.3   188.8         ohc

   num-of-cylinders  horsepower  average-mileage    price
0              four         111               21  13495.0
1              four         111               21  16500.0
2               six         154               19  16500.0
3              four         102               24  13950.0
4              five         115               18  17450.0
..              ...         ...              ...      ...
56             four          85               27   7975.0
57             four          52               37   7995.0
58             four         100               26   9995.0
59             four         114               23  12940.0
60             four         114               23  13415.0

[61 rows x 10 columns]
```

[3]:
```
#1.3 Print the entire dataset in Pandas data frame
df= pd.DataFrame(df)
print(df)
```

```
    index      company   body-style  wheel-base  length engine-type  \
0       0  alfa-romero  convertible        88.6   168.8        dohc
1       1  alfa-romero  convertible        88.6   168.8        dohc
2       2  alfa-romero    hatchback        94.5   171.2        ohcv
3       3         audi        sedan        99.8   176.6         ohc
4       4         audi        sedan        99.4   176.6         ohc
..    ...          ...          ...         ...     ...         ...
56     81   volkswagen        sedan        97.3   171.7         ohc
57     82   volkswagen        sedan        97.3   171.7         ohc
58     86   volkswagen        sedan        97.3   171.7         ohc
59     87        volvo        sedan       104.3   188.8         ohc
60     88        volvo        wagon       104.3   188.8         ohc
```

```
   num-of-cylinders  horsepower  average-mileage    price
0              four         111               21  13495.0
1              four         111               21  16500.0
2               six         154               19  16500.0
3              four         102               24  13950.0
4              five         115               18  17450.0
..              …           …                 …      …
56             four          85               27   7975.0
57             four          52               37   7995.0
58             four         100               26   9995.0
59             four         114               23  12940.0
60             four         114               23  13415.0

[61 rows x 10 columns]
```

[4]:
```
#1.4 Print datatype of all the columns of dataset
df.info()

df.head(5)
df.tail(5)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 61 entries, 0 to 60
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   index             61 non-null     int64
 1   company           61 non-null     object
 2   body-style        61 non-null     object
 3   wheel-base        61 non-null     float64
 4   length            61 non-null     float64
 5   engine-type       61 non-null     object
 6   num-of-cylinders  61 non-null     object
 7   horsepower        61 non-null     int64
 8   average-mileage   61 non-null     int64
 9   price             58 non-null     float64
dtypes: float64(3), int64(3), object(4)
memory usage: 4.9+ KB
```

[4]:
```
    index     company body-style  wheel-base  length engine-type  \
56     81  volkswagen      sedan        97.3   171.7         ohc
57     82  volkswagen      sedan        97.3   171.7         ohc
58     86  volkswagen      sedan        97.3   171.7         ohc
59     87       volvo      sedan       104.3   188.8         ohc
60     88       volvo      wagon       104.3   188.8         ohc

   num-of-cylinders  horsepower  average-mileage    price
```

```
56              four          85               27   7975.0
57              four          52               37   7995.0
58              four         100               26   9995.0
59              four         114               23  12940.0
60              four         114               23  13415.0
```

[5]:
```
#2.        Get a statistical summary of the dataset
#2.1 Display all the statistical indicators of your dataset

df.describe()
```

[5]:
```
            index  wheel-base      length  horsepower  average-mileage  \
count   61.000000   61.000000   61.000000   61.000000        61.000000
mean    40.885246   98.481967  173.098361  107.852459        25.803279
std     25.429706    6.679234   14.021846   53.524398         8.129821
min      0.000000   88.400000  141.100000   48.000000        13.000000
25%     18.000000   94.500000  159.100000   68.000000        19.000000
50%     39.000000   96.300000  171.200000  100.000000        25.000000
75%     61.000000  101.200000  177.300000  123.000000        31.000000
max     88.000000  120.900000  208.100000  288.000000        47.000000

              price
count     58.000000
mean   15387.000000
std    11320.259841
min     5151.000000
25%     6808.500000
50%    11095.000000
75%    18120.500000
max    45400.000000
```

[6]:
```
#2.2 Print the name of all unique columns

df['horsepower'].unique()
df['engine-type'].unique()
```

[6]:
```
array(['dohc', 'ohcv', 'ohc', 'l', 'rotor', 'ohcf', 'dohcv'], dtype=object)
```

[7]:
```
#2.3 Print the count of all unique columns
df['horsepower'].nunique()
df['engine-type'].nunique()
```

[7]:
```
7
```

[8]:
```
#3.        Get a subset of the entire dataset
#3.1 Create a new Pandas dataframe and copy ith rows and jth columns of your␣
 ↪dataset
```

```
df1=df.iloc[2:8,3]
print(df1)
```

```
2     94.5
3     99.8
4     99.4
5     99.8
6    105.8
7    101.2
Name: wheel-base, dtype: float64
```

[9]:
```
df2=df.iloc[2,2:6]
print(df2)
```

```
body-style     hatchback
wheel-base          94.5
length             171.2
engine-type         ohcv
Name: 2, dtype: object
```

[10]:
```
#4 Modify the dataset

mdf=df.rename(columns={'company':'company_name'},inplace=False)
print(mdf)


mdf.rename(columns={'index':'sno'},inplace=True)   #change colum name
print(mdf)
```

```
    index company_name   body-style  wheel-base  length engine-type  \
0       0  alfa-romero  convertible        88.6   168.8        dohc
1       1  alfa-romero  convertible        88.6   168.8        dohc
2       2  alfa-romero    hatchback        94.5   171.2        ohcv
3       3         audi        sedan        99.8   176.6         ohc
4       4         audi        sedan        99.4   176.6         ohc
..    ...          ...          ...         ...     ...         ...
56     81   volkswagen        sedan        97.3   171.7         ohc
57     82   volkswagen        sedan        97.3   171.7         ohc
58     86   volkswagen        sedan        97.3   171.7         ohc
59     87        volvo        sedan       104.3   188.8         ohc
60     88        volvo        wagon       104.3   188.8         ohc

   num-of-cylinders  horsepower  average-mileage    price
0              four         111               21  13495.0
1              four         111               21  16500.0
2               six         154               19  16500.0
3              four         102               24  13950.0
```

```
4              five         115               18  17450.0
..              …          …                …      …
56             four          85               27   7975.0
57             four          52               37   7995.0
58             four         100               26   9995.0
59             four         114               23  12940.0
60             four         114               23  13415.0

[61 rows x 10 columns]
    sno company_name   body-style  wheel-base  length engine-type  \
0     0  alfa-romero  convertible        88.6   168.8        dohc
1     1  alfa-romero  convertible        88.6   168.8        dohc
2     2  alfa-romero    hatchback        94.5   171.2        ohcv
3     3         audi        sedan        99.8   176.6         ohc
4     4         audi        sedan        99.4   176.6         ohc
..  …          …          …          …      …         …
56   81   volkswagen        sedan        97.3   171.7         ohc
57   82   volkswagen        sedan        97.3   171.7         ohc
58   86   volkswagen        sedan        97.3   171.7         ohc
59   87        volvo        sedan       104.3   188.8         ohc
60   88        volvo        wagon       104.3   188.8         ohc

   num-of-cylinders  horsepower  average-mileage    price
0              four         111               21  13495.0
1              four         111               21  16500.0
2               six         154               19  16500.0
3              four         102               24  13950.0
4              five         115               18  17450.0
..              …          …                …      …
56             four          85               27   7975.0
57             four          52               37   7995.0
58             four         100               26   9995.0
59             four         114               23  12940.0
60             four         114               23  13415.0

[61 rows x 10 columns]
```

[14]:
```
#5.        Identify and deal with missing values
#5.1 Display all the tuples with status of missing values

df.isnull()
```

[14]:
```
    index  company  body-style  wheel-base  length  engine-type  \
0   False    False       False       False   False        False
1   False    False       False       False   False        False
2   False    False       False       False   False        False
3   False    False       False       False   False        False
```

```
4   False    False       False       False   False        False
..    …       …          …          …       …            …
56  False    False       False       False   False        False
57  False    False       False       False   False        False
58  False    False       False       False   False        False
59  False    False       False       False   False        False
60  False    False       False       False   False        False

    num-of-cylinders  horsepower  average-mileage  price
0              False       False            False  False
1              False       False            False  False
2              False       False            False  False
3              False       False            False  False
4              False       False            False  False
..               …         …                …      …
56             False       False            False  False
57             False       False            False  False
58             False       False            False  False
59             False       False            False  False
60             False       False            False  False

[61 rows x 10 columns]
```

[16]:
```
#5.2        Count the number of missing values in dataset of columns

pd.isnull(df['price']).sum()
```

[16]:
```
3
```

[17]:
```
#5.3 Print the columns having missing value
df.isnull().sum()
```

[17]:
```
index               0
company             0
body-style          0
wheel-base          0
length              0
engine-type         0
num-of-cylinders    0
horsepower          0
average-mileage     0
price               3
dtype: int64
```

[18]:
```
#5.4 Drop all the rows having atleast one missing values in the same Pandas␣
 ↪dataframe
df.dropna(axis=0,how='any',inplace= True)
```

```
df.isnull().sum()
```

[18]:
```
index                0
company              0
body-style           0
wheel-base           0
length               0
engine-type          0
num-of-cylinders     0
horsepower           0
average-mileage      0
price                0
dtype: int64
```

[19]:
```
#5.5 Drop all the rows where all the values are missing values in the same␣
 ↪Pandas dataframe
df.dropna(axis=0,how='all',inplace= True)
df.isnull().sum()
```

[19]:
```
index                0
company              0
body-style           0
wheel-base           0
length               0
engine-type          0
num-of-cylinders     0
horsepower           0
average-mileage      0
price                0
dtype: int64
```

[20]:
```
#5.6 Drop all the columns having atleast one missing values using different␣
 ↪Pandas dataframe
new_df= df.dropna(axis=1,how='any',inplace=False)
new_df.isnull().sum()
```

[20]:
```
index                0
company              0
body-style           0
wheel-base           0
length               0
engine-type          0
num-of-cylinders     0
horsepower           0
average-mileage      0
price                0
dtype: int64
```

[21]:
```
#5.7 Drop all the columns where all the values are missing values using␣
 ↪different Pandas dataframe

new_df=df.dropna(axis=1,how='all',inplace=False)
new_df.isnull().sum()
```

[21]:
```
index               0
company             0
body-style          0
wheel-base          0
length              0
engine-type         0
num-of-cylinders    0
horsepower          0
average-mileage     0
price               0
dtype: int64
```

[23]:
```
#5.8 Fill the missing value with constant

df['price']=df['price'].fillna(0)
pd.isnull(df['price']).sum()
```

[23]:
```
0
```

[24]:
```
#5.9 Fill the missing value with mean, median and mode

df['horsepower']= pd.to_numeric(df['horsepower'] , errors= 'coerce') #change␣
 ↪datatype object to numeric
df.info()

#mean
df['price']=df['price'].fillna(df['price'].mean())
pd.isnull(df['price']).sum()

#median
df['horsepower']= df['horsepower'].fillna(df['price'].median())
pd.isnull(df['horsepower']).sum()

#mode
df['price']=df['price'].fillna(df['price'].mode()[0])
pd.isnull(df['price']).sum()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 58 entries, 0 to 60
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype
```

```
---  ------             --------------  -----
 0   index              58 non-null     int64
 1   company            58 non-null     object
 2   body-style         58 non-null     object
 3   wheel-base         58 non-null     float64
 4   length             58 non-null     float64
 5   engine-type        58 non-null     object
 6   num-of-cylinders   58 non-null     object
 7   horsepower         58 non-null     int64
 8   average-mileage    58 non-null     int64
 9   price              58 non-null     float64
dtypes: float64(3), int64(3), object(4)
memory usage: 5.0+ KB
```

[24]: 0

[25]:
```
#5.10 Fill the missing value with forward fill, backward fill and interpolate

#forward fill
df['price'].fillna(method="ffill", inplace=True)
pd.isnull(df['price']).sum()

#Backward fill
df['horsepower'].fillna(method="bfill", inplace=True)
pd.isnull(df['horsepower']).sum()

#Interpolate

df['horsepower'].interpolate(method='linear',inplace=True)
pd.isnull(df['horsepower']).sum()

df['price'].interpolate(method='linear',inplace=True)
pd.isnull(df['price']).sum()
```

[25]: 0

[ ]: