1)Question

```
import pandas as pd
df = pd.read_csv("/content/train.csv")
```

Double-click (or enter) to edit

```
missing_values = df.isnull().sum()
```

```
new_df = df.dropna()
new_df = df.drop(columns=['New_Price'])
```

```
new_df.to_csv("Clean.csv", index=False)
```

2) Question

```
# Converting 'Mileage' column into string
df['Mileage'] = df['Mileage'].astype(str)

# to remove "kmpl" from the 'Mileage'
df['Mileage'] = df['Mileage'].str.replace(' kmpl', '')

# Handle missing or empty values by converting them to NaN
df['Mileage'] = df['Mileage'].replace('', None)

# Remove non-numeric characters from 'Mileage'
df['Mileage'] = df['Mileage'].str.replace(' km/kg', '', regex=True)

# Convert to float
df['Mileage'] = df['Mileage'].astype(float)
```

```
df['Engine'] = df['Engine'].astype(str)
df['Engine'] = df['Engine'].str.replace(' CC', '').astype(float)
```

```
df['Power'] = df['Power'].astype(str)
df['Power'] = df['Power'].str.replace(' bhp', '', regex=True)
df['Power'] = df['Power'].str.replace('null', '0', regex=True).astype(float)
```

```
# Convert the 'New_Price' column to a string type
df['New_Price'] = df['New_Price'].astype(str)

# Now, you can use the .str accessor for string operations
df['New_Price'] = df['New_Price'].str.replace('Lakh', '').str.replace('Cr', '', regex=True)

# Handle empty strings by converting them to 0
df['New_Price'] = df['New_Price'].replace('', '0')

# Now, the 'New_Price' column should be processed correctly
print(df)
```

```
      Unnamed: 0                              Name    Location  Year  \
0              1  Hyundai Creta 1.6 CRDi SX Option        Pune  2015
1              2                      Honda Jazz V     Chennai  2011
2              3                 Maruti Ertiga VDI     Chennai  2012
3              4   Audi A4 New 2.0 TDI Multitronic  Coimbatore  2013
4              6            Nissan Micra Diesel XV      Jaipur  2013
...          ...                               ...         ...   ...
5842        6014                  Maruti Swift VDI       Delhi  2014
```

```
5843        6015          Hyundai Xcent 1.1 CRDi S      Jaipur  2015
5844        6016             Mahindra Xylo D4 BSIV      Jaipur  2012
5845        6017                Maruti Wagon R VXI     Kolkata  2013
5846        6018             Chevrolet Beat Diesel   Hyderabad  2011

      Kilometers_Driven Fuel_Type Transmission Owner_Type  Mileage  Engine  \
0                 41000    Diesel       Manual      First    19.67  1582.0
1                 46000    Petrol       Manual      First    13.00  1199.0
2                 87000    Diesel       Manual      First    20.77  1248.0
3                 40670    Diesel    Automatic     Second    15.20  1968.0
4                 86999    Diesel       Manual      First    23.08  1461.0
...                 ...       ...          ...        ...      ...     ...
5842              27365    Diesel       Manual      First    28.40  1248.0
5843             100000    Diesel       Manual      First    24.40  1120.0
5844              55000    Diesel       Manual     Second    14.00  2498.0
5845              46000    Petrol       Manual      First    18.90   998.0
5846              47000    Diesel       Manual      First    25.44   936.0

       Power  Seats New_Price  Price
0     126.20    5.0       nan  12.50
1      88.70    5.0     8.61    4.50
2      88.76    7.0       nan   6.00
3     140.80    5.0       nan  17.74
4      63.10    5.0       nan   3.50
...      ...    ...       ...    ...
5842   74.00    5.0     7.88    4.75
5843   71.00    5.0       nan   4.00
5844  112.00    8.0       nan   2.90
5845   67.10    5.0       nan   2.65
5846   57.60    5.0       nan   2.50

[5847 rows x 14 columns]
```

```
df.to_csv("Updated_data.csv", index=False)
```

3) Question

```
# Convert categorical variables to one-hot encoded values
df = pd.get_dummies(df, columns=['Fuel_Type', 'Transmission'], prefix=['Fuel_Type', 'Transmission'])



df.to_csv("Modified_data.csv", index=False)
```

4) Question

```
import pandas as pd
import datetime

# Calculate the current year
current_year = datetime.datetime.now().year

# Create a new column for the car's age
df['Car_Age'] = current_year - df['Year']

# Print the updated DataFrame
print(df)
```

```
      Unnamed: 0                              Name    Location  Year  \
0              1  Hyundai Creta 1.6 CRDi SX Option        Pune  2015
1              2                      Honda Jazz V     Chennai  2011
2              3                 Maruti Ertiga VDI     Chennai  2012
3              4   Audi A4 New 2.0 TDI Multitronic  Coimbatore  2013
4              6            Nissan Micra Diesel XV      Jaipur  2013
...          ...                               ...         ...   ...
5842        6014                  Maruti Swift VDI       Delhi  2014
5843        6015          Hyundai Xcent 1.1 CRDi S      Jaipur  2015
5844        6016             Mahindra Xylo D4 BSIV      Jaipur  2012
5845        6017                Maruti Wagon R VXI     Kolkata  2013
5846        6018             Chevrolet Beat Diesel   Hyderabad  2011

      Kilometers_Driven Owner_Type  Mileage  Engine   Power  Seats New_Price  \
0                 41000      First    19.67  1582.0  126.20    5.0       nan
1                 46000      First    13.00  1199.0   88.70    5.0     8.61
2                 87000      First    20.77  1248.0   88.76    7.0       nan
3                 40670     Second    15.20  1968.0  140.80    5.0       nan
4                 86999      First    23.08  1461.0   63.10    5.0       nan
...                 ...        ...      ...     ...     ...    ...       ...
5842              27365      First    28.40  1248.0   74.00    5.0     7.88
5843             100000      First    24.40  1120.0   71.00    5.0       nan
5844              55000     Second    14.00  2498.0  112.00    8.0       nan
5845              46000      First    18.90   998.0   67.10    5.0       nan
5846              47000      First    25.44   936.0   57.60    5.0       nan
```

```
      Price  Fuel_Type_Diesel  Fuel_Type_Electric  Fuel_Type_Petrol  \
0     12.50                 1                   0                 0
1      4.50                 0                   0                 1
2      6.00                 1                   0                 0
3     17.74                 1                   0                 0
4      3.50                 1                   0                 0
...     ...               ...                 ...               ...
5842   4.75                 1                   0                 0
5843   4.00                 1                   0                 0
5844   2.90                 1                   0                 0
5845   2.65                 0                   0                 1
5846   2.50                 1                   0                 0

      Transmission_Automatic  Transmission_Manual  Car_Age
0                          0                   1        8
1                          0                   1       12
2                          0                   1       11
3                          1                   0       10
4                          0                   1       10
...                      ...                 ...      ...
5842                       0                   1        9
5843                       0                   1        8
5844                       0                   1       11
5845                       0                   1       10
5846                       0                   1       12

[5847 rows x 18 columns]
```

```
df.to_csv("Final_data.csv", index=False)
```