preprocessing

November 17, 2025

0.1 Preprocessing

```python
[1]: %load_ext autoreload
     %autoreload 2
```

```python
[2]: from pathlib import Path
     import zipfile
     import os
     from enum import Enum

     import pandas as pd
     import kaggle
```

```python
[3]: def download(competition: str, dir: Path | str) -> tuple[Path, Path]:
         """
         Downloads dataset from kaggle competition.

         Args:
           competition (str): Kaggle competition to download dataset from.
           dir (Path | str): Path to download the dataset at.

         Returns:
           (Path, Path): Tuple of training dataset and test dataset paths.
         """
         if isinstance(dir, str):
             dir = Path(dir)
         if not dir.exists():
             dir.mkdir(parents=True)

         kaggle.api.authenticate()
         kaggle.api.competition_download_files(competition, dir)
         zip_file_path = Path(dir, competition).with_suffix(".zip")
         with zipfile.ZipFile(zip_file_path, "r") as zip_ref:
             zip_ref.extractall(dir)
         os.remove(zip_file_path)

         return (dir / "train.csv", dir / "test.csv")
```

```
[4]: train_data_path, test_data_path = download("gist-mldl-25f-hw3", "../dataset")
```

```
[5]: df_train = pd.read_csv(train_data_path)
     df_test = pd.read_csv(test_data_path)
```

```
[6]: df_train
```

```
[6]:           ID                    Name    Location  Year  Kilometers_Driven  \
     0      G4XLU0             Tata Indigo  Coimbatore  2013              59138
     1      CRSHOS          Toyota Corolla       Kochi  2013              81504
     2      FUJ4X1               Ford Ikon   Hyderabad  2007              92000
     3      QMVK6E             Hyundai i20     Kolkata  2012              33249
     4      4SWHFC              Honda City   Bangalore  2011              65000
     ...       ...                     ...         ...   ...                ...
     4465   TR7SLB          Mahindra XUV500       Kochi  2016              51884
     4466   QB41QE              Honda Jazz     Kolkata  2016              27210
     4467   ODG8N7        Land Rover Range        Pune  2015              52000
     4468   EV2ZBX             Maruti Alto       Delhi  2013              56000
     4469   J2RCU8  Mercedes-Benz GL-Class   Bangalore  2014              52000

           Fuel_Type Transmission Owner_Type      Mileage    Engine       Power  \
     0        Diesel       Manual      First   17.0 kmpl  1405 CC       70 bhp
     1        Diesel       Manual      First  21.43 kmpl  1364 CC     87.2 bhp
     2        Petrol       Manual      First   13.8 kmpl  1299 CC       70 bhp
     3        Diesel       Manual      First  21.27 kmpl  1396 CC    88.76 bhp
     4        Petrol       Manual      First   17.0 kmpl  1497 CC      118 bhp
     ...         ...          ...        ...         ...      ...         ...
     4465     Diesel       Manual      First   16.0 kmpl  2179 CC      140 bhp
     4466     Diesel       Manual      First   27.3 kmpl  1498 CC     98.6 bhp
     4467     Diesel    Automatic      First   12.7 kmpl  2179 CC    187.7 bhp
     4468     Petrol       Manual      First   24.7 kmpl   796 CC     47.3 bhp
     4469     Diesel    Automatic      First   12.0 kmpl  2987 CC      224 bhp

                  Colour Seats No. of Doors New_Price  Price
     0            Others     5            4        \N   2.58
     1            Others     5            4        \N   6.53
     2            Others     5            4        \N   1.25
     3      Black/Silver     5            4        \N   3.25
     4             White     5            4        \N   5.20
     ...             ...   ...          ...       ...    ...
     4465          White     7            5        \N  12.46
     4466         Others     5            4        \N   5.85
     4467          White     5            4        \N  39.75
     4468         Others     5            4        \N   2.10
     4469   Black/Silver     7            5        \N  49.00

     [4470 rows x 16 columns]
```
```

```
[7]: df = pd.concat([df_train, df_test])
     df
```

```
[7]:           ID                Name    Location  Year  Kilometers_Driven  Fuel_Type  \
     0      G4XLU0        Tata Indigo  Coimbatore  2013              59138     Diesel
     1      CRSHOS      Toyota Corolla       Kochi  2013              81504     Diesel
     2      FUJ4X1           Ford Ikon   Hyderabad  2007              92000     Petrol
     3      QMVK6E          Hyundai i20     Kolkata  2012              33249     Diesel
     4      4SWHFC          Honda City   Bangalore  2011              65000     Petrol
     ...       ...                ...         ...   ...                ...        ...
     1486   CWRWOT         Tata Safari   Bangalore  2011              80000     Diesel
     1487   Q7Z939  Volkswagen Passat     Kolkata  2011              42500     Diesel
     1488   73KOPC             Audi A4   Bangalore  2014              37600     Diesel
     1489   XEBBLO     Mahindra Scorpio  Bangalore  2011              73000     Diesel
     1490   LOLVST         Hyundai i20  Coimbatore  2017              14618     Petrol

          Transmission Owner_Type      Mileage   Engine       Power        Colour  \
     0          Manual      First   17.0 kmpl  1405 CC       70 bhp        Others
     1          Manual      First  21.43 kmpl  1364 CC     87.2 bhp        Others
     2          Manual      First   13.8 kmpl  1299 CC       70 bhp        Others
     3          Manual      First  21.27 kmpl  1396 CC    88.76 bhp  Black/Silver
     4          Manual      First   17.0 kmpl  1497 CC      118 bhp         White
     ...           ...        ...         ...      ...         ...           ...
     1486       Manual      First  13.93 kmpl  2179 CC   138.03 bhp        Others
     1487    Automatic      First  18.33 kmpl  1968 CC    167.7 bhp  Black/Silver
     1488    Automatic     Second  16.55 kmpl  1968 CC   147.51 bhp  Black/Silver
     1489       Manual      First  12.05 kmpl  2179 CC      120 bhp        Others
     1490       Manual      First   18.6 kmpl  1197 CC    81.83 bhp  Black/Silver

          Seats No. of Doors New_Price  Price
     0        5            4        \N   2.58
     1        5            4        \N   6.53
     2        5            4        \N   1.25
     3        5            4        \N   3.25
     4        5            4        \N   5.20
     ...    ...          ...       ...    ...
     1486     7            5        \N    NaN
     1487     5            4        \N    NaN
     1488     5            4        \N    NaN
     1489     8            5        \N    NaN
     1490     5            4        \N    NaN

     [5961 rows x 16 columns]
```

1. ID

Unique identifier for each car listing

```
[8]: df.ID.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, 0 to 1490
Series name: ID
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 93.1+ KB
```

```
[9]: len(df.ID.unique()) == len(df.ID)
```

```
[9]: True
```

ID feature has no missing value and all unique.

```
[10]: df.set_index("ID", inplace=True)
      df
```

```
[10]:                     Name     Location  Year Kilometers_Driven Fuel_Type  \
      ID
      G4XLU0          Tata Indigo  Coimbatore  2013             59138    Diesel
      CRSHOS        Toyota Corolla       Kochi  2013             81504    Diesel
      FUJ4X1            Ford Ikon   Hyderabad  2007             92000    Petrol
      QMVK6E           Hyundai i20     Kolkata  2012             33249    Diesel
      4SWHFC            Honda City   Bangalore  2011             65000    Petrol
      ...                     ...         ...   ...               ...       ...
      CWRWOT           Tata Safari   Bangalore  2011             80000    Diesel
      Q7Z939    Volkswagen Passat     Kolkata  2011             42500    Diesel
      73K0PC               Audi A4   Bangalore  2014             37600    Diesel
      XEBBL0      Mahindra Scorpio   Bangalore  2011             73000    Diesel
      LOLVST           Hyundai i20  Coimbatore  2017             14618    Petrol

              Transmission Owner_Type     Mileage   Engine      Power        Colour  \
      ID
      G4XLU0        Manual      First   17.0 kmpl  1405 CC     70 bhp        Others
      CRSHOS        Manual      First  21.43 kmpl  1364 CC   87.2 bhp        Others
      FUJ4X1        Manual      First   13.8 kmpl  1299 CC     70 bhp        Others
      QMVK6E        Manual      First  21.27 kmpl  1396 CC  88.76 bhp  Black/Silver
      4SWHFC        Manual      First   17.0 kmpl  1497 CC    118 bhp         White
      ...              ...        ...         ...      ...        ...           ...
      CWRWOT        Manual      First  13.93 kmpl  2179 CC 138.03 bhp        Others
      Q7Z939     Automatic      First  18.33 kmpl  1968 CC   167.7 bhp  Black/Silver
      73K0PC     Automatic     Second  16.55 kmpl  1968 CC 147.51 bhp  Black/Silver
      XEBBL0        Manual      First  12.05 kmpl  2179 CC    120 bhp        Others
      LOLVST        Manual      First   18.6 kmpl  1197 CC  81.83 bhp  Black/Silver
```

```
       Seats No. of Doors New_Price  Price
ID
G4XLU0     5            4        \N   2.58
CRSHOS     5            4        \N   6.53
FUJ4X1     5            4        \N   1.25
QMVK6E     5            4        \N   3.25
4SWHFC     5            4        \N   5.20
...       ...          ...      ...  ...
CWRWOT     7            5        \N   NaN
Q7Z939     5            4        \N   NaN
73K0PC     5            4        \N   NaN
XEBBL0     8            5        \N   NaN
LOLVST     5            4        \N   NaN

[5961 rows x 15 columns]
```

2. Name

The brand and model name of the car (e.g. Hundai i20, Honda City)

[11]:
```python
df.Name.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: Name
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 93.1+ KB
```

[12]:
```python
df.Name.unique()
```

[12]:
```
array(['Tata Indigo', 'Toyota Corolla', 'Ford Ikon', 'Hyundai i20',
       'Honda City', 'Ford Ecosport', 'Hyundai Grand', 'Maruti Wagon',
       'Mercedes-Benz GLA', 'Jaguar XF', 'Porsche Cayenne', 'BMW 3',
       'Mercedes-Benz New', 'Tata Manza', 'Fiat Linea', 'Maruti Swift',
       'Mercedes-Benz GLE', 'BMW 5', 'Ford Fiesta', 'Honda Accord',
       'Maruti Alto', 'Mahindra XUV500', 'Fiat Petra', 'Skoda Laura',
       'Maruti Baleno', 'Jeep Compass', 'BMW X1', 'Hyundai EON',
       'Ford Figo', 'Hyundai i10', 'Toyota Innova', 'Renault Duster',
       'Skoda Superb', 'Toyota Etios', 'Hyundai Verna', 'Honda WRV',
       'Mahindra Scorpio', 'Maruti Esteem', 'Nissan Sunny',
       'Nissan Terrano', 'Audi Q3', 'Ford EcoSport', 'BMW Z4',
       'Maruti Dzire', 'BMW X5', 'Audi Q7', 'Honda Amaze',
       'Mercedes-Benz E-Class', 'Volkswagen Polo', 'Tata Indica',
       'Chevrolet Cruze', 'Maruti Ertiga', 'Chevrolet Spark',
       'Mercedes-Benz A', 'Maruti Eeco', 'Honda Brio', 'Ford Endeavour',
```

```
'Mercedes-Benz M-Class', 'Hyundai Creta', 'Volkswagen Vento',
'Hyundai Xcent', 'Audi A7', 'Mercedes-Benz CLA', 'Skoda Octavia',
'Chevrolet Captiva', 'Tata New', 'Force One', 'Honda Jazz',
'Mahindra Bolero', 'BMW X3', 'Jaguar F', 'Skoda Fabia',
'Mitsubishi Cedia', 'Tata Xenon', 'Maruti Ritz', 'BMW 7',
'Mahindra Xylo', 'Maruti Vitara', 'Maruti Zen', 'Toyota Fortuner',
'Mahindra Renault', 'Hyundai Elantra', 'Fiat Siena',
'Honda Mobilio', 'Chevrolet Beat', 'Mahindra Ssangyong',
'Tata Safari', 'Renault KWID', 'Mercedes-Benz GLS', 'Honda Civic',
'Volkswagen Ameo', 'Maruti 800', 'Audi A4', 'Chevrolet Enjoy',
'Honda CR-V', 'Hyundai Accent', 'Maruti Grand', 'Skoda Rapid',
'Tata Nano', 'Mercedes-Benz B', 'Audi Q5', 'Honda BRV',
'Land Rover Range', 'Mahindra KUV', 'Volkswagen Passat',
'Maruti Ignis', 'Renault Captur', 'Datsun redi-GO', 'Jaguar XJ',
'Maruti SX4', 'Mercedes-Benz GL-Class', 'Maruti Ciaz',
'Maruti Celerio', 'Mahindra XUV300', 'Mahindra TUV',
'Hyundai Santro', 'Tata Zest', 'Mercedes-Benz GLC',
'Volkswagen Jetta', 'Datsun Redi', 'Chevrolet Optra',
'Maruti Omni', 'Maruti A-Star', 'Audi A6', 'Maruti S',
'Mini Cooper', 'Mahindra Logan', 'Chevrolet Aveo',
'Mercedes-Benz S', 'Hyundai Santa', 'Mercedes-Benz C-Class',
'Honda BR-V', 'Tata Sumo', 'Smart Fortwo', 'Fiat Grande',
'Mitsubishi Montero', 'Chevrolet Sail', 'Audi RS5',
'Porsche Panamera', 'Land Rover Discovery', 'Mahindra Thar',
'Tata Nexon', 'Mahindra Quanto', 'Tata Tiago', 'Mitsubishi Pajero',
'Isuzu MUX', 'Land Rover Freelander', 'Ford Fusion', 'Mahindra E',
'Skoda Yeti', 'Hyundai Tucson', 'Mahindra Verito', 'Datsun GO',
'Nissan Micra', 'Renault Fluence', 'Renault Pulse',
'Mercedes-Benz R-Class', 'Volvo XC60', 'BMW 6', 'Tata Tigor',
'BMW X6', 'Volvo S60', 'Mercedes-Benz S-Class', 'Toyota Camry',
'Nissan X-Trail', 'Ford Aspire', 'Ford Freestyle', 'Tata Bolt',
'ISUZU D-MAX', 'Toyota Qualis', 'Porsche Boxster', 'Hyundai Elite',
'Hyundai Getz', 'Fiat Punto', 'Hyundai Sonata', 'Porsche Cayman',
'Jaguar XE', 'Audi A8', 'Maruti S-Cross', 'Fiat Avventura',
'Volkswagen CrossPolo', 'Toyota Prius', 'Volvo V40',
'Renault Scala', 'Bentley Continental', 'Tata Hexa', 'Audi A3',
'Mahindra Jeep', 'Mitsubishi Lancer', 'Mahindra NuvoSport',
'Renault Koleos', 'BMW 1', 'Volvo S80', 'Mini Clubman',
'Mercedes-Benz SLK-Class', 'Toyota Platinum', 'Mini Countryman',
'Volkswagen Beetle', 'Nissan Evalia', 'Mercedes-Benz SLC',
'Audi TT', 'Nissan Teana', 'Mercedes-Benz CLS-Class',
'Lamborghini Gallardo', 'Honda WR-V', 'Mitsubishi Outlander',
'Volvo XC90', 'Mercedes-Benz SL-Class', 'Ford Classic',
'Volkswagen Tiguan', 'Ford Mustang', 'Maruti 1000'], dtype=object)
```

Name encodes both the brand and model. Since the brand itself is considered to be a critical feature, separate Name into Brand and Model. Note that Brand and Model has a hirarchical relationship. Not all of the cartesian product of Brand and Model is valid.

Naive string splitting won't work due to edge cases like `Land Rover Range` and `Mahindra Ssangyong`.

```python
[13]: class Brand(Enum):
          """
          Enum of recognized brands. All the values are in title-case. Transform to
          title-case before comparing.
          """

          Audi = "Audi"
          Bentley = "Bentley"
          BMW = "Bmw"
          Chevrolet = "Chevrolet"
          Datsun = "Datsun"
          Fiat = "Fiat"
          Force = "Force"
          Ford = "Ford"
          Honda = "Honda"
          Hyundai = "Hyundai"
          Isuzu = "Isuzu"
          Jaguar = "Jaguar"
          Jeep = "Jeep"
          Lamborghini = "Lamborghini"
          Land_Rover = "Land Rover"
          Mahindra = "Mahindra"
          Maruti = "Maruti"
          Mercedes_Benz = "Mercedes-Benz"
          Mini = "Mini"
          Mitsubishi = "Mitsubishi"
          Nissan = "Nissan"
          Porsche = "Porsche"
          Renault = "Renault"
          Skoda = "Skoda"
          Smart = "Smart"
          Tata = "Tata"
          Toyota = "Toyota"
          Volkswagen = "Volkswagen"
          Volvo = "Volvo"
```

```python
[14]: for name in df.Name.unique():
          brands = tuple(brand.value for brand in set(Brand))
          if not name.title().startswith(brands):
              print(f"Could not match any brand in the name: {name}")
              break
      else:
          print("Every brand name recognized!")
```

```
Every brand name recognized!
```

```python
[15]: def split_brand_and_model_from_name(series: pd.Series) -> tuple[pd.Series, pd.
      ⤷Series]:
          """
          From a given series of car names, it splits the name into brand and model.

          Args:
              series (Series): A series of names to be processed.

          Returns:
              tuple[Series, Series]: A tuple of Brand series and Model series.
          """
          names = series.str.title()
          brands = pd.Series([None] * len(names), index=names.index, dtype=object)
          models = pd.Series([None] * len(names), index=names.index, dtype=object)

          for brand in Brand:
              condition = names.str.startswith(brand.value, na=False) & brands.isna()

              if condition.any():
                  brands.loc[condition] = brand.name
                  matched_names = names.loc[condition]
                  residuals = matched_names.str[len(brand.value) :].str.strip()
                  models.loc[condition] = residuals.where(residuals != "", None)

          brands = brands.rename("Brand").astype("category")
          models = models.rename("Model").astype("category")

          return brands, models
```

```python
[16]: df["Brand"], df["Model"] = split_brand_and_model_from_name(df.Name)
      df
```

```
[16]:                      Name      Location   Year  Kilometers_Driven  Fuel_Type  \
      ID
      G4XLU0         Tata Indigo    Coimbatore  2013              59138     Diesel
      CRSHOS       Toyota Corolla        Kochi  2013              81504     Diesel
      FUJ4X1            Ford Ikon    Hyderabad  2007              92000     Petrol
      QMVK6E          Hyundai i20      Kolkata  2012              33249     Diesel
      4SWHFC           Honda City    Bangalore  2011              65000     Petrol
      ...                    ...           ...   ...                ...        ...
      CWRWOT          Tata Safari    Bangalore  2011              80000     Diesel
      Q7Z939    Volkswagen Passat      Kolkata  2011              42500     Diesel
      73KOPC              Audi A4    Bangalore  2014              37600     Diesel
      XEBBLO     Mahindra Scorpio    Bangalore  2011              73000     Diesel
      LOLVST          Hyundai i20    Coimbatore  2017              14618     Petrol

                Transmission  Owner_Type     Mileage    Engine        Power         Colour  \
```

```
            ID
G4XLU0        Manual        First    17.0 kmpl  1405 CC      70 bhp        Others
CRSHOS        Manual        First   21.43 kmpl  1364 CC    87.2 bhp        Others
FUJ4X1        Manual        First    13.8 kmpl  1299 CC      70 bhp        Others
QMVK6E        Manual        First   21.27 kmpl  1396 CC   88.76 bhp  Black/Silver
4SWHFC        Manual        First    17.0 kmpl  1497 CC     118 bhp         White
...             ...          ...          ...      ...         ...           ...
CWRWOT        Manual        First   13.93 kmpl  2179 CC  138.03 bhp        Others
Q7Z939     Automatic        First   18.33 kmpl  1968 CC   167.7 bhp  Black/Silver
73KOPC     Automatic       Second   16.55 kmpl  1968 CC  147.51 bhp  Black/Silver
XEBBLO        Manual        First   12.05 kmpl  2179 CC     120 bhp        Others
LOLVST        Manual        First    18.6 kmpl  1197 CC   81.83 bhp  Black/Silver

           Seats  No. of Doors New_Price  Price        Brand    Model
ID
G4XLU0         5             4        \N   2.58         Tata   Indigo
CRSHOS         5             4        \N   6.53       Toyota  Corolla
FUJ4X1         5             4        \N   1.25         Ford     Ikon
QMVK6E         5             4        \N   3.25      Hyundai      I20
4SWHFC         5             4        \N   5.20        Honda     City
...          ...           ...       ...   ...          ...      ...
CWRWOT         7             5        \N   NaN         Tata   Safari
Q7Z939         5             4        \N   NaN   Volkswagen   Passat
73KOPC         5             4        \N   NaN         Audi       A4
XEBBLO         8             5        \N   NaN     Mahindra  Scorpio
LOLVST         5             4        \N   NaN      Hyundai      I20

[5961 rows x 17 columns]
```

[17]:
```python
groupby_brand = df.groupby("Brand", observed=True)
brand_histogram = groupby_brand.nunique()["Model"].sort_values(ascending=False)
brand_histogram
```

[17]:
```
Brand
Maruti          22
Mercedes_Benz   19
Mahindra        16
Hyundai         15
Tata            14
Honda           12
Ford            10
BMW             10
Audi            10
Toyota           8
Chevrolet        8
Volkswagen       8
Renault          7
```

```
       Fiat          6
       Skoda         6
       Nissan        6
       Mitsubishi    5
       Volvo         5
       Porsche       4
       Jaguar        4
       Mini          3
       Datsun        3
       Land_Rover    3
       Isuzu         2
       Lamborghini   1
       Jeep          1
       Smart         1
       Force         1
       Bentley       1
       Name: Model, dtype: int64
```

```python
[18]: with pd.option_context(
          "display.max_rows",
          None,
      ):
          display(df[["Brand", "Model"]].groupby(["Brand", "Model"], observed=True).
      ↪count())
```

```
Empty DataFrame
Columns: []
```

```
Index: [(Audi, A3), (Audi, A4), (Audi, A6), (Audi, A7), (Audi, A8), (Audi, Q3),␣
↪(Audi, Q5), (Audi, Q7), (Audi, Rs5), (Audi, Tt), (BMW, 1), (BMW, 3), (BMW, 5),␣
↪(BMW, 6), (BMW, 7), (BMW, X1), (BMW, X3), (BMW, X5), (BMW, X6), (BMW, Z4),␣
↪(Bentley, Continental), (Chevrolet, Aveo), (Chevrolet, Beat), (Chevrolet,␣
↪Captiva), (Chevrolet, Cruze), (Chevrolet, Enjoy), (Chevrolet, Optra),␣
↪(Chevrolet, Sail), (Chevrolet, Spark), (Datsun, Go), (Datsun, Redi), (Datsun,␣
↪Redi-Go), (Fiat, Avventura), (Fiat, Grande), (Fiat, Linea), (Fiat, Petra),␣
↪(Fiat, Punto), (Fiat, Siena), (Force, One), (Ford, Aspire), (Ford, Classic),␣
↪(Ford, Ecosport), (Ford, Endeavour), (Ford, Fiesta), (Ford, Figo), (Ford,␣
↪Freestyle), (Ford, Fusion), (Ford, Ikon), (Ford, Mustang), (Honda, Accord),␣
↪(Honda, Amaze), (Honda, Br-V), (Honda, Brio), (Honda, Brv), (Honda, City),␣
↪(Honda, Civic), (Honda, Cr-V), (Honda, Jazz), (Honda, Mobilio), (Honda, Wr-V),␣
↪(Honda, Wrv), (Hyundai, Accent), (Hyundai, Creta), (Hyundai, Elantra),␣
↪(Hyundai, Elite), (Hyundai, Eon), (Hyundai, Getz), (Hyundai, Grand), (Hyundai,␣
↪I10), (Hyundai, I20), (Hyundai, Santa), (Hyundai, Santro), (Hyundai, Sonata),␣
↪(Hyundai, Tucson), (Hyundai, Verna), (Hyundai, Xcent), (Isuzu, D-Max), (Isuzu,␣
↪Mux), (Jaguar, F), (Jaguar, Xe), (Jaguar, Xf), (Jaguar, Xj), (Jeep, Compass),␣
↪(Lamborghini, Gallardo), (Land_Rover, Discovery), (Land_Rover, Freelander),␣
↪(Land_Rover, Range), (Mahindra, Bolero), (Mahindra, E), (Mahindra, Jeep),␣
↪(Mahindra, Kuv), (Mahindra, Logan), (Mahindra, Nuvosport), (Mahindra, Quanto),␣
↪(Mahindra, Renault), (Mahindra, Scorpio), (Mahindra, Ssangyong), (Mahindra,␣
↪Thar), (Mahindra, Tuv), (Mahindra, Verito), …]
```

```
[19]: df.drop(columns=["Name"], inplace=True)
```

3. Location

The city where the car is being sold

```
[20]: df.Location.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: Location
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 222.2+ KB
```

```
[21]: df.Location.unique()
```

```
[21]: array(['Coimbatore', 'Kochi', 'Hyderabad', 'Kolkata', 'Bangalore',
             'Delhi', 'Pune', 'Chennai', 'Mumbai', 'Ahmedabad', 'Jaipur', '\\N'],
            dtype=object)
```

These seem to be locations in India. Some missing values are denoted with \\N.

```
[22]: df.Location = df.Location.replace("\\N", None).astype("category")
      df.Location.unique()
```

```
[22]: ['Coimbatore', 'Kochi', 'Hyderabad', 'Kolkata', 'Bangalore', …, 'Chennai',
       'Mumbai', 'Ahmedabad', 'Jaipur', NaN]
      Length: 12
      Categories (11, object): ['Ahmedabad', 'Bangalore', 'Chennai', 'Coimbatore',
      …, 'Kochi', 'Kolkata', 'Mumbai', 'Pune']
```

```
[23]: df.Location.value_counts()
```

```
[23]: Location
      Mumbai        781
      Hyderabad     739
      Kochi         646
      Coimbatore    630
      Pune          611
      Delhi         549
      Kolkata       526
      Chennai       489
      Jaipur        406
      Bangalore     351
      Ahmedabad     222
      Name: count, dtype: int64
```

There seems to be no location with extremely small sample size.

4. Year

Manufacturing year of the vehicle

```
[24]: df.Year.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: Year
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 222.2+ KB
```

```
[25]: df.Year.unique()
```

```
[25]: array(['2013', '2007', '2012', '2011', '2014', '2016', '2019', '2015',
             '2008', '2010', '2017', '2005', '2009', '2018', '2004', '2006',
             '2001', '1999', '2002', '2003', '2000', '\\N', '1998', 2012, 2008,
             2010, 2017, 2014, 2011, 2015, 2018, 2016, 2009, 2013, 2004, 2007,
             2019, 1998, 2005, 2000, 2006, 2003, 2002, 2001], dtype=object)
```

Similarly, missing values are represented with \\N.

```
[26]: df.Year = df.Year.replace("\\N", None)
      df.Year.unique()
```

```
[26]: array(['2013', '2007', '2012', '2011', '2014', '2016', '2019', '2015',
             '2008', '2010', '2017', '2005', '2009', '2018', '2004', '2006',
             '2001', '1999', '2002', '2003', '2000', None, '1998', 2012, 2008,
             2010, 2017, 2014, 2011, 2015, 2018, 2016, 2009, 2013, 2004, 2007,
             2019, 1998, 2005, 2000, 2006, 2003, 2002, 2001], dtype=object)
```

```
[27]: pd.to_numeric(df.Year, errors="coerce").astype("Int64").value_counts().
      ↪sort_index()
```

```
[27]: Year
      1998      4
      1999      2
      2000      4
      2001      7
      2002     14
      2003     13
      2004     28
      2005     55
      2006     75
      2007    123
      2008    170
      2009    196
      2010    338
      2011    461
      2012    573
      2013    642
      2014    793
      2015    736
      2016    740
      2017    586
      2018    298
      2019    101
      Name: count, dtype: Int64
```

There are some years with significantly small sample size.

Setting the year 2020 as the current year, converting year to `Age` might be more intuitive value relevant to `New_Price`.

```
[28]: current_year = 2020

      df["Age"] = current_year - pd.to_numeric(df.Year, errors="coerce").
      ↪astype("Int64")
```

```
df.Age.value_counts(sort=False).sort_index()
```

[28]: Age
1   101
2   298
3   586
4   740
5   736
6   793
7   642
8   573
9   461
10   338
11   196
12   170
13   123
14   75
15   55
16   28
17   13
18   14
19   7
20   4
21   2
22   4
Name: count, dtype: Int64

5. Kilometers Driven

Total distance the car has been driven (in kilometers)

[29]:
```
df.Kilometers_Driven.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: Kilometers_Driven
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 222.2+ KB
```

[30]:
```
df.Kilometers_Driven = df.Kilometers_Driven.replace("\\N", None)
df[df.Kilometers_Driven.isna()]
```

[30]:   Location Year Kilometers_Driven Fuel_Type Transmission Owner_Type \
ID
VA5F28   NaN 2017   None   Petrol   Manual   First

```

|        |        |      |      |        |           |        |
|--------|--------|------|------|--------|-----------|--------|
| BVPJHJ | NaN    | 2013 | None | Diesel | Automatic | Second |
| 5ZGUKG | NaN    | 2018 | None | Diesel | Manual    | First  |
| LWTYCE | NaN    | 2013 | None | Diesel | Automatic | First  |
| HZTZU8 | NaN    | 2014 | None | Petrol | Manual    | Second |
| CZ59WU | NaN    | 2010 | None | Diesel | Automatic | First  |
| EGWEU4 | NaN    | 2009 | None | Petrol | Manual    | Second |
| MGGIOB | NaN    | 2012 | None | Petrol | Manual    | First  |

|        | Mileage    | Engine  | Power     | Colour | Seats | No. of Doors | New_Price  |
|--------|------------|---------|-----------|--------|-------|--------------|------------|
| ID     |            |         |           |        |       |              |            |
| VA5F28 | 18.15 kmpl | 1198 CC | 82 bhp    | Others | 5     | 4            | \N         |
| BVPJHJ | 11.18 kmpl | 2696 CC | 184 bhp   | Others | 7     | 5            | \N         |
| 5ZGUKG | 24.3 kmpl  | 1248 CC | 88.5 bhp  | White  | 5     | 4            | 11.12 Lakh |
| LWTYCE | 11.74 kmpl | 2987 CC | 254.8 bhp | Others | 5     | 4            | \N         |
| HZTZU8 | 16.09 kmpl | 1598 CC | 103.5 bhp | White  | 5     | 4            | 12.33 Lakh |
| CZ59WU | 12.4 kmpl  | 2698 CC | 179.5 bhp | Others | 5     | 4            | \N         |
| EGWEU4 | 14.0 kmpl  | 1061 CC | 64 bhp    | White  | 5     | 4            | \N         |
| MGGIOB | 14.53 kmpl | 1798 CC | 138.1 bhp | White  | 5     | 4            | \N         |

|        | Price | Brand         | Model     | Age |
|--------|-------|---------------|-----------|-----|
| ID     |       |               |           |     |
| VA5F28 | 4.85  | Mahindra      | Kuv       | 3   |
| BVPJHJ | 12.50 | Mahindra      | Ssangyong | 7   |
| 5ZGUKG | 8.63  | Maruti        | Vitara    | 2   |
| LWTYCE | 28.00 | Mercedes_Benz | M-Class   | 7   |
| HZTZU8 | 6.98  | Volkswagen    | Vento     | 6   |
| CZ59WU | NaN   | Audi          | A6        | 10  |
| EGWEU4 | NaN   | Maruti        | Wagon     | 11  |
| MGGIOB | NaN   | Toyota        | Corolla   | 8   |

[31]:
```python
df.Kilometers_Driven = pd.to_numeric(df.Kilometers_Driven)
df.Kilometers_Driven = df.Kilometers_Driven.astype("Int64")
df.Kilometers_Driven
```

[31]:
```
ID
G4XLU0    59138
CRSHOS    81504
FUJ4X1    92000
QMVK6E    33249
4SWHFC    65000
          ...
CWRW0T    80000
Q7Z939    42500
73K0PC    37600
XEBBL0    73000
LOLVST    14618
Name: Kilometers_Driven, Length: 5961, dtype: Int64
```

6. Fuel_Type

Type of fuel the car uses (Pertrol, Diesel, CNG, LPG, Electric)

```
[32]: df.Fuel_Type.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: Fuel_Type
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 222.2+ KB
```

```
[33]: df.Fuel_Type.unique()
```

```
[33]: array(['Diesel', 'Petrol', 'CNG', 'LPG', 'Electric'], dtype=object)
```

```
[34]: df.Fuel_Type = df.Fuel_Type.astype("category")
      df.Fuel_Type.cat.categories
```

```
[34]: Index(['CNG', 'Diesel', 'Electric', 'LPG', 'Petrol'], dtype='object')
```

```
[35]: df.Fuel_Type.value_counts()
```

```
[35]: Fuel_Type
      Diesel      3188
      Petrol      2705
      CNG           56
      LPG           10
      Electric       2
      Name: count, dtype: int64
```

7. Transmission

Type of transmission system (Manual or Automatic)

```
[36]: df.Transmission.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: Transmission
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 222.2+ KB
```

```
[37]: df.Transmission.unique()
```

```
[37]: array(['Manual', 'Automatic', '\\N'], dtype=object)
```

```
[38]: df.Transmission.value_counts()
```

```
[38]: Transmission
      Manual       4225
      Automatic    1709
      \N             27
      Name: count, dtype: int64
```

```
[39]: df.Transmission = df.Transmission.replace("\\N", None)
      df[df.Transmission.isna()]
```

```
[39]:           Location  Year  Kilometers_Driven Fuel_Type Transmission Owner_Type  \
      ID
      VVH3NN      Chennai  2012              65932    Diesel         None     Second
      HFU1G9       Jaipur  2015             100000    Diesel         None         \N
      43PACK   Coimbatore  2013              40670    Diesel         None     Second
      JRH5YV      Chennai  2011              46000    Petrol         None      First
      FIU4TU    Hyderabad  2013              40000    Diesel         None         \N
      6FJFYS    Hyderabad  2012              75000       LPG         None      First
      9FLMYR         Pune  2015              41000    Diesel         None      First
      BIVO6Q   Coimbatore  2015              70602    Diesel         None         \N
      GWE95I       Mumbai  2010              72000       CNG         None      First
      S149E7        Delhi  2011              68000    Diesel         None         \N
      ABRKHB   Coimbatore  2007              66800    Petrol         None         \N
      HIJ508      Chennai  2012              87000    Diesel         None      First
      9Z2LPT       Mumbai  2009             102002    Diesel         None         \N
      VRWP96       Mumbai  2016              36000    Diesel         None      First
      MTRG04    Bangalore  2015              67600    Petrol         None         \N
      N2LG5N        Kochi  2018              25692    Petrol         None      First
      7NKZMQ         Pune  2016              37208    Diesel         None         \N
      3WVXX0   Coimbatore  2011              45004    Diesel         None         \N
      HCYSXM    Hyderabad  2009              53000    Petrol         None         \N
      6FH9L9        Delhi  2014              27365    Diesel         None         \N
      QWA2Y3       Mumbai  2011              38000    Petrol         None         \N
      V871B7       Jaipur  2013              86999    Diesel         None      First
      MTN68R       Mumbai  2015              33500    Petrol         None         \N
      3D7ZY1         Pune  2013              64430    Diesel         None      First
      YQ2I8J        Delhi  2013              33746    Petrol         None         \N
      G7M8HP      Kolkata  2012              60000    Petrol         None      First
      H37YTM    Bangalore  2012              45000    Petrol         None     Second

                 Mileage   Engine      Power    Colour Seats No. of Doors  \
      ID
      VVH3NN   22.3 kmpl  1248 CC    74 bhp     White     5             4
      HFU1G9   24.4 kmpl  1120 CC    71 bhp    Others     5             4
```

ID	Mileage	Engine	Power	Color		
43PACK	15.2 kmpl	1968 CC	140.8 bhp	White	5	4
JRH5YV	18.2 kmpl	1199 CC	88.7 bhp	White	5	4
FIU4TU	17.85 kmpl	2967 CC	300 bhp	Black/Silver	4	4
6FJFYS	21.1 km/kg	814 CC	55.2 bhp	White	5	4
9FLMYR	19.67 kmpl	1582 CC	126.2 bhp	White	5	4
BIVO6Q	25.8 kmpl	1498 CC	98.6 bhp	Others	5	4
GWE95I	26.6 km/kg	998 CC	58.16 bhp	White	5	4
S149E7	19.3 kmpl	1248 CC	73.9 bhp	Black/Silver	5	4
ABRKHB	15.3 kmpl	1341 CC	83 bhp	Others	5	4
HIJ508	20.77 kmpl	1248 CC	88.76 bhp	Others	7	5
9Z2LPT	8.7 kmpl	2987 CC	224.34 bhp	Others	5	4
VRWP96	11.36 kmpl	2755 CC	171.5 bhp	White	8	5
MTRG04	23.1 kmpl	998 CC	67.04 bhp	Black/Silver	5	4
N2LG5N	21.56 kmpl	1462 CC	103.25 bhp	Others	5	4
7NKZMQ	24.3 kmpl	1248 CC	88.5 bhp	White	5	4
3WVXX0	12.8 kmpl	2494 CC	102 bhp	Others	7	4
HCYSXM	0.0 kmpl	3597 CC	262.6 bhp	White	5	4
6FH9L9	28.4 kmpl	1248 CC	74 bhp	Black/Silver	5	4
QWA2Y3	16.09 kmpl	1598 CC	103.5 bhp	Black/Silver	5	4
V871B7	23.08 kmpl	1461 CC	63.1 bhp	White	5	4
MTN68R	19.16 kmpl	2494 CC	158.2 bhp	Others	5	4
3D7ZY1	20.54 kmpl	1598 CC	103.6 bhp	White	5	4
YQ2I8J	18.5 kmpl	1198 CC	86.8 bhp	White	5	4
G7M8HP	16.8 kmpl	1497 CC	116.3 bhp	White	5	4
H37YTM	19.4 kmpl	1198 CC	86.8 bhp	White	5	4

ID	New_Price	Price	Brand	Model	Age
VVH3NN	\N	1.95	Tata	Indica	8
HFU1G9	\N	4.00	Hyundai	Xcent	5
43PACK	\N	17.74	Audi	A4	7
JRH5YV	8.61 Lakh	4.50	Honda	Jazz	9
FIU4TU	\N	45.00	Porsche	Panamera	7
6FJFYS	\N	2.35	Hyundai	Eon	8
9FLMYR	\N	12.50	Hyundai	Creta	5
BIVO6Q	\N	4.83	Honda	Amaze	5
GWE95I	\N	1.75	Maruti	Wagon	10
S149E7	\N	2.75	Maruti	Swift	9
ABRKHB	\N	2.20	Hyundai	Getz	13
HIJ508	\N	6.00	Maruti	Ertiga	8
9Z2LPT	\N	10.75	Mercedes_Benz	M-Class	11
VRWP96	21 Lakh	17.50	Toyota	Innova	4
MTRG04	\N	4.00	Maruti	Celerio	5
N2LG5N	10.65 Lakh	9.95	Maruti	Ciaz	2
7NKZMQ	9.93 Lakh	7.43	Maruti	Vitara	4
3WVXX0	\N	9.48	Toyota	Innova	9
HCYSXM	\N	NaN	Skoda	Superb	11

```
6FH9L9    7.88 Lakh      NaN      Maruti      Swift    6
QWA2Y3   11.91 Lakh      NaN   Volkswagen     Vento    9
V871B7           \N      NaN       Nissan     Micra    7
MTN68R           \N      NaN       Toyota     Camry    5
3D7ZY1           \N      NaN   Volkswagen     Vento    7
YQ2I8J    6.63 Lakh      NaN        Honda      Brio    7
G7M8HP           \N      NaN        Honda      City    8
H37YTM           \N      NaN        Honda      Brio    8
```

[40]:
```python
df.Transmission = df.Transmission.astype("category")
df.Transmission.cat.categories
```

[40]: Index(['Automatic', 'Manual'], dtype='object')

8. Owner_Type

Number of previous owners

[41]:
```python
df.Owner_Type.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: Owner_Type
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 222.2+ KB
```

[42]:
```python
df.Owner_Type.unique()
```

[42]: array(['First', 'Second', 'Third', '\\N', 'Fourth & Above'], dtype=object)

[43]:
```python
df.Owner_Type = df.Owner_Type.replace("\\N", None)
df[df.Owner_Type.isna()]
```

[43]:
ID	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type
2BGQXK	Delhi	2012	62000	Diesel	Manual	None
HFU1G9	Jaipur	2015	100000	Diesel	NaN	None
FIU4TU	Hyderabad	2013	40000	Diesel	NaN	None
BIVO6Q	Coimbatore	2015	70602	Diesel	NaN	None
S149E7	Delhi	2011	68000	Diesel	NaN	None
ABRKHB	Coimbatore	2007	66800	Petrol	NaN	None
9Z2LPT	Mumbai	2009	102002	Diesel	NaN	None
MTRG04	Bangalore	2015	67600	Petrol	NaN	None
7NKZMQ	Pune	2016	37208	Diesel	NaN	None
3WVXX0	Coimbatore	2011	45004	Diesel	NaN	None

```
HCYSXM    Hyderabad  2009            53000    Petrol         NaN      None
6FH9L9        Delhi  2014            27365    Diesel         NaN      None
QWA2Y3       Mumbai  2011            38000    Petrol         NaN      None
MTN68R       Mumbai  2015            33500    Petrol         NaN      None
YQ2I8J        Delhi  2013            33746    Petrol         NaN      None


            Mileage    Engine       Power      Colour Seats  No. of Doors  \
ID
2BGQXK   22.32 kmpl   1582 CC  126.32 bhp       White     5             4
HFU1G9    24.4 kmpl   1120 CC      71 bhp      Others     5             4
FIU4TU   17.85 kmpl   2967 CC     300 bhp  Black/Silver   4             4
BIVO6Q    25.8 kmpl   1498 CC    98.6 bhp      Others     5             4
S149E7    19.3 kmpl   1248 CC    73.9 bhp  Black/Silver   5             4
ABRKHB    15.3 kmpl   1341 CC      83 bhp      Others     5             4
9Z2LPT     8.7 kmpl   2987 CC  224.34 bhp      Others     5             4
MTRG04    23.1 kmpl    998 CC   67.04 bhp  Black/Silver   5             4
7NKZMQ    24.3 kmpl   1248 CC    88.5 bhp       White     5             4
3WVXX0    12.8 kmpl   2494 CC     102 bhp      Others     7             4
HCYSXM     0.0 kmpl   3597 CC   262.6 bhp       White     5             4
6FH9L9    28.4 kmpl   1248 CC      74 bhp  Black/Silver   5             4
QWA2Y3   16.09 kmpl   1598 CC   103.5 bhp  Black/Silver   5             4
MTN68R   19.16 kmpl   2494 CC   158.2 bhp      Others     5             4
YQ2I8J    18.5 kmpl   1198 CC    86.8 bhp       White     5             4


            New_Price  Price          Brand      Model  Age
ID
2BGQXK            \N    4.60        Hyundai      Verna    8
HFU1G9            \N    4.00        Hyundai      Xcent    5
FIU4TU            \N   45.00        Porsche   Panamera    7
BIVO6Q            \N    4.83          Honda      Amaze    5
S149E7            \N    2.75         Maruti      Swift    9
ABRKHB            \N    2.20        Hyundai       Getz   13
9Z2LPT            \N   10.75  Mercedes_Benz    M-Class   11
MTRG04            \N    4.00         Maruti    Celerio    5
7NKZMQ     9.93 Lakh    7.43         Maruti     Vitara    4
3WVXX0            \N    9.48         Toyota     Innova    9
HCYSXM            \N     NaN          Skoda     Superb   11
6FH9L9     7.88 Lakh     NaN         Maruti      Swift    6
QWA2Y3    11.91 Lakh     NaN     Volkswagen      Vento    9
MTN68R            \N     NaN         Toyota      Camry    5
YQ2I8J     6.63 Lakh     NaN          Honda       Brio    7
```

Owner_Type is ordinal category. It should be orderable.

```python
owner_type_order = {"First": 1, "Second": 2, "Third": 3, "Fourth & Above": 4}

ordinal_owner_category = pd.CategoricalDtype(
```

```
        categories=list(owner_type_order.keys()), ordered=True
    )
df.Owner_Type = df.Owner_Type.astype(ordinal_owner_category)
df.Owner_Type.cat.categories
```

[44]: Index(['First', 'Second', 'Third', 'Fourth & Above'], dtype='object')

9. Mileage

Fuel efficiency of the car (kmpl or km/kg)

[45]: ```
df.Mileage.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: Mileage
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 222.2+ KB
```

[46]: ```
df.Mileage.head()
```

[46]: ```
ID
G4XLU0   17.0 kmpl
CRSHOS   21.43 kmpl
FUJ4X1   13.8 kmpl
QMVK6E   21.27 kmpl
4SWHFC   17.0 kmpl
Name: Mileage, dtype: object
```

[47]: ```
df.Mileage = df.Mileage.replace("\\N", None)
```

[48]: ```
df[["_Mileage_Value", "_Mileage_Unit"]] = df.Mileage.str.split(expand=True)
df._Mileage_Unit.unique()
```

[48]: array(['kmpl', 'km/kg', None], dtype=object)

[49]: ```
df._Mileage_Value = df._Mileage_Value.astype("float64")
df._Mileage_Unit = df._Mileage_Unit.astype("category")
```

kmpl and km/kg units are mixed.

kmpl and km/kg would have the same scale if the density was 1 (water). However, depending on the fuel type, the density can be different, thus the conversion factor too.

Luckily, Fuel_Type is available.

[50]: ```
df.Fuel_Type.cat.categories
```

```
[50]: Index(['CNG', 'Diesel', 'Electric', 'LPG', 'Petrol'], dtype='object')
```

```
[51]: conversion_factors = {"CNG": 1.33, "Diesel": 1.20, "LPG": 1.85, "Petrol": 1.35}

      df.Mileage = df._Mileage_Value.copy()

      for fuel_type, factor in conversion_factors.items():
          mask = (df._Mileage_Unit == "km/kg") & (df.Fuel_Type == fuel_type)
          df.loc[mask, "Mileage"] = df.loc[mask, "_Mileage_Value"] * factor
      df = df.drop(["_Mileage_Unit", "_Mileage_Value"], axis=1)

      df.Mileage
```

```
[51]: ID
      G4XLU0      17.00
      CRSHOS      21.43
      FUJ4X1      13.80
      QMVK6E      21.27
      4SWHFC      17.00
                   …
      CWRW0T      13.93
      Q7Z939      18.33
      73K0PC      16.55
      XEBBL0      12.05
      LOLVST      18.60
      Name: Mileage, Length: 5961, dtype: float64
```

10. Engine

Engine displacement in cubic centimeters (CC)

```
[52]: df.Engine.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: Engine
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 222.2+ KB
```

```
[53]: df.Engine.head()
```

```
[53]: ID
      G4XLU0      1405 CC
      CRSHOS      1364 CC
      FUJ4X1      1299 CC
```

```
QMVK6E    1396 CC
4SWHFC    1497 CC
Name: Engine, dtype: object
```

[54]: `df.Engine = df.Engine.replace("\\N", None)`

[55]:
```python
df.Engine = pd.to_numeric(df.Engine.str.split(expand=True)[0])
df.Engine = df.Engine.astype("Int64")
df.Engine
```

[55]:
```
ID
G4XLU0    1405
CRSHOS    1364
FUJ4X1    1299
QMVK6E    1396
4SWHFC    1497

          …
CWRW0T    2179
Q7Z939    1968
73K0PC    1968
XEBBL0    2179
LOLVST    1197
Name: Engine, Length: 5961, dtype: Int64
```

11. Power

Maximum power output of the engine (bhp - brake horsepower)

[56]: `df.Power.info()`

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: Power
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 222.2+ KB
```

[57]: `df.Power = df.Power.replace("\\N", None)`

[58]:
```python
df.Power = pd.to_numeric(df.Power.str.split(expand=True)[0], errors="coerce")
df.Power
```

[58]:
```
ID
G4XLU0    70.00
CRSHOS    87.20
FUJ4X1    70.00
```

```
QMVK6E      88.76
4SWHFC     118.00
              …
CWRW0T     138.03
Q7Z939     167.70
73K0PC     147.51
XEBBL0     120.00
LOLVST      81.83
Name: Power, Length: 5961, dtype: float64
```

12. Colour

Exterior color of the vehicle

```
[59]:  df.Colour.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: Colour
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 222.2+ KB
```

```
[60]:  df.Colour.head()
```

```
[60]:  ID
       G4XLU0          Others
       CRSHOS          Others
       FUJ4X1          Others
       QMVK6E     Black/Silver
       4SWHFC           White
       Name: Colour, dtype: object
```

```
[61]:  df.Colour.unique()
```

```
[61]:  array(['Others', 'Black/Silver', 'White', '\\N'], dtype=object)
```

```
[62]:  df.Colour = df.Colour.replace("\\N", None)
```

```
[63]:  df.Colour = df.Colour.astype("category")
       df.Colour
```

```
[63]:  ID
       G4XLU0          Others
       CRSHOS          Others
       FUJ4X1          Others
```

```
QMVK6E      Black/Silver
4SWHFC            White
              …
CWRWOT            Others
Q7Z939      Black/Silver
73KOPC      Black/Silver
XEBBLO            Others
LOLVST      Black/Silver
Name: Colour, Length: 5961, dtype: category
Categories (3, object): ['Black/Silver', 'Others', 'White']
```

13. Seats

Number of seating capacity

```
[64]: df.Seats.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: Seats
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 222.2+ KB
```

```
[65]: df.Seats.head()
```

```
[65]: ID
      G4XLU0    5
      CRSHOS    5
      FUJ4X1    5
      QMVK6E    5
      4SWHFC    5
      Name: Seats, dtype: object
```

```
[66]: df.Seats = df.Seats.replace("\\N", None)
```

```
[67]: df.Seats = pd.to_numeric(df.Seats, errors="coerce")
      df.Seats = df.Seats.astype("Int64")
      df.Seats
```

```
[67]: ID
      G4XLU0    5
      CRSHOS    5
      FUJ4X1    5
      QMVK6E    5
      4SWHFC    5
```

```
               ..
CWRWOT      7
Q7Z939      5
73KOPC      5
XEBBLO      8
LOLVST      5
Name: Seats, Length: 5961, dtype: Int64
```

14. No. of Doors

Number of doors in the vehicle

```
[68]: df["No. of Doors"].info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLUO to LOLVST
Series name: No. of Doors
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 222.2+ KB
```

```
[69]: df.rename(columns={"No. of Doors": "Doors"}, inplace=True)
```

```
[70]: df.Doors
```

```
[70]: ID
G4XLUO      4
CRSHOS      4
FUJ4X1      4
QMVK6E      4
4SWHFC      4
               ..
CWRWOT      5
Q7Z939      4
73KOPC      4
XEBBLO      5
LOLVST      4
Name: Doors, Length: 5961, dtype: object
```

```
[71]: df.Doors.unique()
```

```
[71]: array(['4', '5', '\\N', '2', 4, 5, 2], dtype=object)
```

```
[72]: df.Doors = df.Doors.replace("\\N", None)
```

```
[73]: df.Doors = pd.to_numeric(df.Doors)
      df.Doors = df.Doors.astype("Int64")
      df.Doors
```

```
[73]: ID
      G4XLU0    4
      CRSHOS    4
      FUJ4X1    4
      QMVK6E    4
      4SWHFC    4
               ..
      CWRW0T    5
      Q7Z939    4
      73K0PC    4
      XEBBL0    5
      LOLVST    4
      Name: Doors, Length: 5961, dtype: Int64
```

15. New_Price

Original price of the car when it was new (may contain missing values)

```
[74]: df.New_Price.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: New_Price
Non-Null Count  Dtype
--------------  -----
5961 non-null   object
dtypes: object(1)
memory usage: 222.2+ KB
```

```
[75]: df.New_Price = df.New_Price.replace("\\N", None)
      df[df.New_Price.isna()]
```

```
[75]:          Location  Year  Kilometers_Driven  Fuel_Type  Transmission Owner_Type  \
      ID
      G4XLU0  Coimbatore  2013              59138     Diesel        Manual      First
      CRSHOS       Kochi  2013              81504     Diesel        Manual      First
      FUJ4X1   Hyderabad  2007              92000     Petrol        Manual      First
      QMVK6E     Kolkata  2012              33249     Diesel        Manual      First
      4SWHFC   Bangalore  2011              65000     Petrol        Manual      First
      ...            ...   ...                ...        ...           ...        ...
      CWRW0T   Bangalore  2011              80000     Diesel        Manual      First
      Q7Z939     Kolkata  2011              42500     Diesel     Automatic      First
      73K0PC   Bangalore  2014              37600     Diesel     Automatic     Second
      XEBBL0   Bangalore  2011              73000     Diesel        Manual      First
```

```
       LOLVST  Coimbatore  2017              14618  Petrol       Manual       First

               Mileage  Engine   Power        Colour  Seats  Doors New_Price  Price  \
       ID
       G4XLU0   17.00    1405   70.00        Others      5      4      None   2.58
       CRSHOS   21.43    1364   87.20        Others      5      4      None   6.53
       FUJ4X1   13.80    1299   70.00        Others      5      4      None   1.25
       QMVK6E   21.27    1396   88.76  Black/Silver      5      4      None   3.25
       4SWHFC   17.00    1497  118.00         White      5      4      None   5.20
       ...        ...     ...     ...           ...    ...    ...       ...    ...
       CWRW0T   13.93    2179  138.03        Others      7      5      None    NaN
       Q7Z939   18.33    1968  167.70  Black/Silver      5      4      None    NaN
       73K0PC   16.55    1968  147.51  Black/Silver      5      4      None    NaN
       XEBBL0   12.05    2179  120.00        Others      8      5      None    NaN
       LOLVST   18.60    1197   81.83  Black/Silver      5      4      None    NaN

                    Brand    Model  Age
       ID
       G4XLU0        Tata   Indigo    7
       CRSHOS      Toyota  Corolla    7
       FUJ4X1        Ford     Ikon   13
       QMVK6E     Hyundai      I20    8
       4SWHFC       Honda     City    9
       ...            ...      ...  ...
       CWRW0T        Tata   Safari    9
       Q7Z939  Volkswagen   Passat    9
       73K0PC        Audi       A4    6
       XEBBL0    Mahindra  Scorpio    9
       LOLVST     Hyundai      I20    3

       [5137 rows x 17 columns]
```

5137 out of 5961 samples are missing `New_Price`.

```
[76]: df.New_Price[df.New_Price.notna()]
```

```
[76]: ID
      B0UPCL   79.43 Lakh
      2FFBR0   21.72 Lakh
      90EINM    8.17 Lakh
      QU3AAV   95.13 Lakh
      4QPAO1   33.36 Lakh
                   ...
      74N7UN    8.82 Lakh
      RKGVCP    8.27 Lakh
      SX4HME   13.72 Lakh
      V7CA4M   15.94 Lakh
      GDLPH2    7.85 Lakh
```

```
Name: New_Price, Length: 824, dtype: object
```

[77]:
```python
df[["_New_Price_Value", "_New_Price_Unit"]] = df.New_Price.str.
 ↪split(expand=True)
df._New_Price_Value = pd.to_numeric(df._New_Price_Value)
df._New_Price_Unit = df._New_Price_Unit.astype("category")
```

[78]:
```python
df._New_Price_Unit.cat.categories
```

[78]:
```
Index(['Cr', 'Lakh'], dtype='object')
```

Lakh and Cr unit is mixed. Standardizing to Lakh. 1 Cr is 100 lakh.

[79]:
```python
cr_mask = df._New_Price_Unit == "Cr"
df.loc[cr_mask, "_New_Price_Value"] = df.loc[cr_mask, "_New_Price_Value"] * 100
df.loc[cr_mask, "_New_Price_Unit"] = "Lakh"
```

[80]:
```python
df.New_Price = df._New_Price_Value.copy()
df.New_Price[df.New_Price.notna()]
```

[80]:
```
ID
BOUPCL    79.43
2FFBR0    21.72
90EINM     8.17
QU3AAV    95.13
4QPAO1    33.36
          ...
74N7UN     8.82
RKGVCP     8.27
SX4HME    13.72
V7CA4M    15.94
GDLPH2     7.85
Name: New_Price, Length: 824, dtype: float64
```

[81]:
```python
df.drop(columns=["_New_Price_Value", "_New_Price_Unit"], inplace=True)
```

16 Price

Target variable: Current selling price of the used car (in lakhs - 1 lakh = 100,000 Indian Rupees)

[82]:
```python
df.Price.info()
```

```
<class 'pandas.core.series.Series'>
Index: 5961 entries, G4XLU0 to LOLVST
Series name: Price
Non-Null Count  Dtype
--------------  -----
4470 non-null   float64
```

```
dtypes: float64(1)
memory usage: 222.2+ KB
```

[83]: `df.Price`

[83]:
```
ID
G4XLU0     2.58
CRSHOS     6.53
FUJ4X1     1.25
QMVK6E     3.25
4SWHFC     5.20
            …
CWRWOT      NaN
Q7Z939      NaN
73K0PC      NaN
XEBBL0      NaN
LOLVST      NaN
Name: Price, Length: 5961, dtype: float64
```

0.1.1 Impute

[84]:
```python
df_train = df.loc[df_train.ID, :]
df_test = df.loc[df_test.ID, :]
```

[85]:
```python
print("# of missing values in df_train for each features (in %)")
for feature in df_train.columns:
    missing_ratio = len(df_train[df_train[feature].isna()]) / len(df_train) *␣
 ↪100
    print(
        f"{feature.ljust(max(len(f) + 1 for f in df_train.columns))}: {'{:.2f}'.
 ↪format(missing_ratio).rjust(6)}% {'<-' if missing_ratio > 1.0 else ''}"
    )

print("\n# of missing values in df_test for each features (in %)")
for feature in df_test.columns:
    missing_ratio = len(df_test[df_test[feature].isna()]) / len(df_test) * 100
    print(
        f"{feature.ljust(max(len(f) + 1 for f in df_test.columns))}: {'{:.2f}'.
 ↪format(missing_ratio).rjust(6)}% {'<-' if missing_ratio > 1.0 else ''}"
    )
```

```
# of missing values in df_train for each features (in %)
Location          :    0.18%
Year              :    0.04%
Kilometers_Driven :    0.11%
Fuel_Type         :    0.00%
```

```
Transmission       :    0.40%
Owner_Type         :    0.22%
Mileage            :    0.04%
Engine             :    0.25%
Power              :    2.37% <-
Colour             :    0.18%
Seats              :    0.04%
Doors              :    0.02%
New_Price          :   86.31% <-
Price              :    0.00%
Brand              :    0.00%
Model              :    0.00%
Age                :    0.04%

# of missing values in df_test for each features (in %)
Location           :    0.20%
Year               :    0.00%
Kilometers_Driven  :    0.20%
Fuel_Type          :    0.00%
Transmission       :    0.60%
Owner_Type         :    0.34%
Mileage            :    0.00%
Engine             :    0.40%
Power              :    1.95% <-
Colour             :    0.20%
Seats              :    0.20%
Doors              :    0.00%
New_Price          :   85.78% <-
Price              :  100.00% <-
Brand              :    0.00%
Model              :    0.00%
Age                :    0.00%
```

Features except `Power` and `New_Price` has less than 1% missing values, and thus rows with missing features other than `Power` or `New_Price` can be safely dropped.

[86]:
```python
print(f"# of training data samples before dropping: {len(df_train)}")
df_train.dropna(
    subset=list(set(df_train.columns) - {"Power", "New_Price"}), inplace=True
)
print(f"# of training data samples after  dropping: {len(df_train)}")

df_train
```

```
# of training data samples before dropping: 4470
# of training data samples after  dropping: 4428
```

[86]:

	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type
ID						
G4XLU0	Coimbatore	2013	59138	Diesel	Manual	First
CRSHOS	Kochi	2013	81504	Diesel	Manual	First
FUJ4X1	Hyderabad	2007	92000	Petrol	Manual	First
QMVK6E	Kolkata	2012	33249	Diesel	Manual	First
4SWHFC	Bangalore	2011	65000	Petrol	Manual	First
…	…	…	…	…	…	…
TR7SLB	Kochi	2016	51884	Diesel	Manual	First
QB41QE	Kolkata	2016	27210	Diesel	Manual	First
ODG8N7	Pune	2015	52000	Diesel	Automatic	First
EV2ZBX	Delhi	2013	56000	Petrol	Manual	First
J2RCU8	Bangalore	2014	52000	Diesel	Automatic	First

	Mileage	Engine	Power	Colour	Seats	Doors	New_Price	Price
ID								
G4XLU0	17.00	1405	70.00	Others	5	4	NaN	2.58
CRSHOS	21.43	1364	87.20	Others	5	4	NaN	6.53
FUJ4X1	13.80	1299	70.00	Others	5	4	NaN	1.25
QMVK6E	21.27	1396	88.76	Black/Silver	5	4	NaN	3.25
4SWHFC	17.00	1497	118.00	White	5	4	NaN	5.20
…	…	…	…	…	…	…	…	…
TR7SLB	16.00	2179	140.00	White	7	5	NaN	12.46
QB41QE	27.30	1498	98.60	Others	5	4	NaN	5.85
ODG8N7	12.70	2179	187.70	White	5	4	NaN	39.75
EV2ZBX	24.70	796	47.30	Others	5	4	NaN	2.10
J2RCU8	12.00	2987	224.00	Black/Silver	7	5	NaN	49.00

	Brand	Model	Age
ID			
G4XLU0	Tata	Indigo	7
CRSHOS	Toyota	Corolla	7
FUJ4X1	Ford	Ikon	13
QMVK6E	Hyundai	I20	8
4SWHFC	Honda	City	9
…	…	…	…
TR7SLB	Mahindra	Xuv500	4
QB41QE	Honda	Jazz	4
ODG8N7	Land_Rover	Range	5
EV2ZBX	Maruti	Alto	7
J2RCU8	Mercedes_Benz	Gl-Class	6

[4428 rows x 17 columns]

[87]: df_test

```
[87]:           Location   Year  Kilometers_Driven Fuel_Type Transmission Owner_Type  \
        ID
        INQOD6        Pune   2012              63400    Diesel       Manual      First
        S7ZJIY     Chennai   2008              89000    Diesel    Automatic     Second
        CZ59WU         NaN   2010               <NA>    Diesel    Automatic      First
        P6II8S      Mumbai   2017              32000    Petrol    Automatic      First
        5OOX2V   Coimbatore  2012              77283    Petrol       Manual      First
        ...           ...   ...                ...       ...          ...        ...
        CWRWOT   Bangalore   2011              80000    Diesel       Manual      First
        Q7Z939     Kolkata   2011              42500    Diesel    Automatic      First
        73KOPC   Bangalore   2014              37600    Diesel    Automatic     Second
        XEBBLO   Bangalore   2011              73000    Diesel       Manual      First
        LOLVST   Coimbatore  2017              14618    Petrol       Manual      First

                Mileage  Engine   Power         Colour  Seats  Doors  New_Price  Price  \
        ID
        INQOD6    17.80    1399   67.00   Black/Silver      5      4        NaN    NaN
        S7ZJIY    16.07    1995  181.00   Black/Silver      4      4        NaN    NaN
        CZ59WU    12.40    2698  179.50         Others      5      4        NaN    NaN
        P6II8S    14.84    1598  103.52   Black/Silver      5      4       13.7    NaN
        5OOX2V    17.00    1497  118.00   Black/Silver      5      4        NaN    NaN
        ...         ...     ...     ...            ...    ...    ...        ...    ...
        CWRWOT    13.93    2179  138.03         Others      7      5        NaN    NaN
        Q7Z939    18.33    1968  167.70   Black/Silver      5      4        NaN    NaN
        73KOPC    16.55    1968  147.51   Black/Silver      5      4        NaN    NaN
        XEBBLO    12.05    2179  120.00         Others      8      5        NaN    NaN
        LOLVST    18.60    1197   81.83   Black/Silver      5      4        NaN    NaN

                    Brand    Model  Age
        ID
        INQOD6       Ford   Fiesta    8
        S7ZJIY        BMW        3   12
        CZ59WU       Audi       A6   10
        P6II8S      Skoda    Rapid    3
        5OOX2V      Honda     City    8
        ...           ...      ...  ...
        CWRWOT       Tata   Safari    9
        Q7Z939  Volkswagen  Passat    9
        73KOPC       Audi       A4    6
        XEBBLO   Mahindra  Scorpio    9
        LOLVST    Hyundai      I20    3

        [1491 rows x 17 columns]
```