الجمهورية اليمنية

جامعة إب

كلية العلوم

قسم علوم الحاسوب وتقنية المعلومات

تكليف مقرر

تنقيب بيانات ـ عملى

Data Mining

المحاضرة الرابعة

عمل الطالب :

أسامة سعيد محمد حمود سعيد ـ مجموعة A

إشراف :

أ. مالك المصنف

2024 - 2025

شرح خطوات ال Data Cleaning :

1 - Include Libraries :

استدعاء مكتبة Pandas لقراءة ملفات CSV و JSON و EXCEL من اجل عملية تنظيف البيانات

2 - Read Data From files :

قراءة البيانات من داخل الملفات لمصفوفات للبدء بتنفيذ العمليات

3 - Know my Dataset :

التعرف على طبيعة البيانات التي سيتم العمل عليها من خلال طباعة بعض القيم بداية ال Data ومعرفة أسماء الاعمدة لكل Dataset

4 - Datasets Cleaning :

البدء بعملية تنظيف البيانات ، الخطوة الأولى كانت تهدف الى توحيد اسماء الاعمدة فقمنا باعاة تسمية الاعمدة لل Dataset الثانية والثالثة حسب ال Dataset الأولى ، بعد ذلك قمنا بحذف الاعمدة Unamed و Random Feature من ال Dataset الأولى والثالثة ، بعد ذلك تحققنا من عدم وجود قيم فارغة NaN في الصفوف للمصفوفات كاملة ، بعد ذلك انتقلنا لتنظيف كل dataset منفردة

5 - Dataset2 Cleaning :

طاعة صفين من محتوى Dataset2 و Dataset1 للتعرف على طبيعة البيانات ، بعد ذلك قمنا بمعالجة الاعمدة لاجل توحيد قيم البيانات لكليهما .

- Fuel Type Column :

تعديل محتوى العمود بحيث يتوافق مع ال Dataset الأخرى وتغيير نوع البيانات

- Paint Type Column :

تعديل محتوى العمود بحيث يتوافق مع ال Dataset الأخرى

- Doors Column :

تعديل محتوى الاعمدة لل Dataset1 و Dataset2 وتعديل نوع البيانات لها

- Weight Column :

تعديل نوع البيانات للعمود بحيث يتوافق مع الاعمدة الأخرى لل Dataset

5 - Dataset3 Cleaning :

طاعة صفين من محتوى Dataset3 و Dataset1 للتعرف على طبيعة البيانات ، بعد ذلك قمنا بمعالجة الاعمدة لاجل توحيد قيم البيانات لكليهما .

- Paint Type Columns :

تعديل محتوى العمود بحيث يتوافق مع ال Dataset الأخرى

6 - Get Shape of Datasets :

بعد الانتهاء من عملية التنظيف قمنا بالتاكد من ابعاد ال Datasets

7 - Drop Columns from Datasets :

حذفنا بعض الاعمدة من ال Dataset بعضها يمكن ان نستنتجها من أعمدة أخرى و العمود Age Group تم حذفها بسبب اختلاف قيم ال metadata لكلاهما وكذلك عدم وجودها في ال Dataset الثالثة

8 - Get Head From Datasets :

طاعة صفين من محتويات ال Datasets للتعرف على طبيعة البيانات

9 - Know Information After Cleaning Datasets :

عرض المعلومات عن ال Datasets الجديدة

10 - Integrate My Datasets :

دمج كل ال Datasets بعد عملية المعالجة

11 - Check my New Dataset :

التحقق من ال Dataset الجديدة بعد الربط حيث ظهرت قيم فارغة في عمود ال Vehicle Age

12 - Clean my New Dataset :

معالجة ال Dataset الجديدة حيث قمنا بعمل الوسط بدلا من القيم الفارغة وحولنا نوع البيانات للعمود

13 - Dataset After Cleaning :

إعادة أسماء الاعمدة الاصلية وعرض ال Dataset بعد عملية المعالجة

14 - Check My New Dataset :

التحقق بعد عملية المعالجة لل Dataset الجديدة

15 - Write Dataset to CSV File :

حفظ ال Dataset المعالجة في ملف CSV

include Library

```
import pandas as pd
```

Reda Datasets

```
dataset1 = pd.read_csv("Toyta1.csv")

dataset1
```

```
     Unnamed: 0  Car_Price  Vehicle_Age  KM_Travelled Fuel_Type
HP  \
0             0      13500           23         46986    Diesel   90

1             1      13750           23         72937    Diesel   90

2             2      13950           24         41711    Diesel   90

3             3      14950           26         48000    Diesel   90

4             4      13750           30         38500    Diesel   90

..          ...        ...          ...           ...       ...  ...

473         473      11950           56         65000    Petrol  110

474         474      10450           48         64193    Petrol  110

475         475       8950           54         64000    Petrol   97

476         476      10250           54         63792    Petrol  110

477         477       9930           53         63635    Petrol  110


       Paint_Type Transmission_Type  Engine_Size  Doors  Weight
Age_Group
0        Metallic            Manual         2000 1165.0 ثلاثة       Old

1        Metallic            Manual         2000 1165.0 ثلاثة       Old

2        Metallic            Manual         2000 1165.0 ثلاثة       Old

3    Non-Metallic            Manual         2000 1165.0 ثلاثة       Old

4    Non-Metallic            Manual         2000 1170.0 ثلاثة       Old
```

```
..            ...               ...          ...    ...     ...
...
473      Metallic            Manual          1600   1075.0 خمسة
Old
474      Metallic            Manual          1600  1040.0 ثلاثة       Old

475      Metallic            Manual          1400  1025.0 ثلاثة       Old

476      Metallic            Manual          1600   1075.0 خمسة
Old
477      Metallic            Manual          1600  1035.0 أربعة
Old

[478 rows x 12 columns]
```

```
dataset2 = pd.read_json("Toyta2.json")

dataset2
```

```
      Cost  Age_in_Years  Total_KM  FuelClass   HP   Body_Color  \
0    10500            54     63135          1  110         Main
1    11950            54     63123          1  110         Main
2    11500            55     63000          0   69         Main
3    11500            55     63000          1  110  Alternative
4    11450            54     62987          1  110  Alternative
..     ...           ...       ...        ...  ...          ...
473   8950            57     52548          1  110  Alternative
474   8400            60     52487          1  110         Main
475   9250            66     52383          1   86  Alternative
476   8900            61     52112          1  110         Main
477   8750            58     51712          1  110  Alternative

    Transmission_Type  Engine_Size  Doors  Weight Price_Category
Age_Group
0              Manual         1600  three    1050         Medium
0
1              Manual         1600   four    1035         Medium
0
2              Manual         1900   five    1140         Medium
0
3              Manual         1600   four    1035         Medium
0
4              Manual         1600   five    1080         Medium
0
..                ...          ...    ...     ...            ...
...
473            Manual         1600  three    1050            Low
0
474            Manual         1600   four    1035            Low
0
```

```
475            Manual        1300  three    1015            Low
0
476            Manual        1600   four    1035            Low
0
477            Manual        1600  three    1050            Low
0

[478 rows x 12 columns]
```

```
dataset3 = pd.read_excel("Toyta3.xlsx")
```

```
dataset3
```

```
     Unnamed: 0.2  Unnamed: 0.1  Unnamed: 0  Sale_Price  Kilometers  \
0               0             0         956       10950       51421   
1               1             1         957        8950       51235   
2               2             2         958        8950       51000   
3               3             3         959        8895       50925   
4               4             4         960        9390       50806   
..            ...           ...         ...         ...         ...   
475           475           475        1431        7500       20544   
476           476           476        1432       10845       11000   
477           477           477        1433        8500       17016   
478           478           478        1434        7250       11000   
479           479           479        1435        6950           1   

     Energy_Source   HP Exterior_Finish Transmission_Type  Engine_Size  
Doors  \
0                1  110       Secondary              Auto         1600   
5   
1                1   86         Primary            Manual         1300   
4   
2                1   86         Primary            Manual         1300   
3   
3                1  110         Primary            Manual         1600   
5   
4                1   86       Secondary            Manual         1300   
3   
..             ...  ...             ...               ...          ...   
...   
475              1   86         Primary            Manual         1300   
3   
476              1   86       Secondary            Manual         1300   
3   
477              1   86       Secondary            Manual         1300   
3   
478              1   86         Primary            Manual         1300   
3   
479              1  110       Secondary            Manual         1600   
5   
```

```
     Weight Price_Category Random_Feature
0      1105         Medium              E
1      1000            Low              B
2      1015            Low              C
3      1070            Low              B
4      1480            Low              D
..      ...            ...            ...
475    1025            Low              C
476    1015         Medium              B
477    1015            Low              B
478    1015            Low              D
479    1114            Low              D

[480 rows x 14 columns]
```

Know My DataSets

Get the Head of Datasets

```
dataset1.head()
```

```
   Unnamed: 0  Car_Price  Vehicle_Age  KM_Travelled Fuel_Type  HP  \
0           0      13500           23         46986    Diesel  90
1           1      13750           23         72937    Diesel  90
2           2      13950           24         41711    Diesel  90
3           3      14950           26         48000    Diesel  90
4           4      13750           30         38500    Diesel  90

     Paint_Type Transmission_Type  Engine_Size  Doors  Weight
Age_Group
0      Metallic            Manual         2000  1165.0 ثلاثة        Old
1      Metallic            Manual         2000  1165.0 ثلاثة        Old
2      Metallic            Manual         2000  1165.0 ثلاثة        Old
3  Non-Metallic            Manual         2000  1165.0 ثلاثة        Old
4  Non-Metallic            Manual         2000  1170.0 ثلاثة        Old
```

```
dataset2.head()
```

```
    Cost  Age_in_Years  Total_KM  FuelClass   HP   Body_Color  \
0  10500            54     63135          1  110         Main
1  11950            54     63123          1  110         Main
2  11500            55     63000          0   69         Main
3  11500            55     63000          1  110  Alternative
4  11450            54     62987          1  110  Alternative

  Transmission_Type  Engine_Size  Doors  Weight Price_Category
Age_Group
```

```
0            Manual         1600  three    1050         Medium
0
1            Manual         1600   four    1035         Medium
0
2            Manual         1900   five    1140         Medium
0
3            Manual         1600   four    1035         Medium
0
4            Manual         1600   five    1080         Medium
0

dataset3.head()

   Unnamed: 0.2  Unnamed: 0.1  Unnamed: 0  Sale_Price  Kilometers  \
0             0             0         956       10950       51421
1             1             1         957        8950       51235
2             2             2         958        8950       51000
3             3             3         959        8895       50925
4             4             4         960        9390       50806

   Energy_Source   HP Exterior_Finish Transmission_Type  Engine_Size
Doors  \
0              1  110       Secondary              Auto         1600
5
1              1   86         Primary            Manual         1300
4
2              1   86         Primary            Manual         1300
3
3              1  110         Primary            Manual         1600
5
4              1   86       Secondary            Manual         1300
3

   Weight Price_Category Random_Feature
0    1105         Medium              E
1    1000            Low              B
2    1015            Low              C
3    1070            Low              B
4    1480            Low              D
```

Get Columns Name

```
dataset1.columns

Index(['Unnamed: 0', 'Car_Price', 'Vehicle_Age', 'KM_Travelled',
'Fuel_Type',
       'HP', 'Paint_Type', 'Transmission_Type', 'Engine_Size',
'Doors',
       'Weight', 'Age_Group'],
      dtype='object')
```

```
dataset2.columns
```

```
Index(['Cost', 'Age_in_Years', 'Total_KM', 'FuelClass', 'HP',
'Body_Color',
       'Transmission_Type', 'Engine_Size', 'Doors', 'Weight',
'Price_Category',
       'Age_Group'],
      dtype='object')
```

```
dataset3.columns
```

```
Index(['Unnamed: 0.2', 'Unnamed: 0.1', 'Unnamed: 0', 'Sale_Price',
       'Kilometers', 'Energy_Source', 'HP', 'Exterior_Finish',
       'Transmission_Type', 'Engine_Size', 'Doors', 'Weight',
'Price_Category',
       'Random_Feature'],
      dtype='object')
```

Datasets Cleaning

Rename Columns For Datasets

```
dataset2.rename(columns =
{'Age_in_Years':'Vehicle_Age','Total_KM':'KM_Travelled','FuelClass':'F
uel_Type','Body_Color':'Paint_Type','Cost':'Car_Price'}, inplace =
True)
```

```
dataset2.columns
```

```
Index(['Car_Price', 'Vehicle_Age', 'KM_Travelled', 'Fuel_Type', 'HP',
       'Paint_Type', 'Transmission_Type', 'Engine_Size', 'Doors',
'Weight',
       'Price_Category', 'Age_Group'],
      dtype='object')
```

```
dataset3.rename(columns =
{'Sale_Price':'Car_Price','Kilometers':'KM_Travelled','Energy_Source':
'Fuel_Type','Exterior_Finish':'Paint_Type'},inplace = True)
```

```
dataset3.columns
```

```
Index(['Unnamed: 0.2', 'Unnamed: 0.1', 'Unnamed: 0', 'Car_Price',
       'KM_Travelled', 'Fuel_Type', 'HP', 'Paint_Type',
'Transmission_Type',
       'Engine_Size', 'Doors', 'Weight', 'Price_Category',
'Random_Feature'],
      dtype='object')
```

Drop Coulmns Unnamed and Random Features from dataset1 and dataset3

```
dataset1.drop(columns = ['Unnamed: 0'] ,inplace =True)
dataset3.drop(columns = ['Unnamed: 0.2','Unnamed: 0.1','Unnamed:
0','Random_Feature'] ,inplace = True)

print (dataset1.columns)
print (dataset3.columns)
```

```
Index(['Car_Price', 'Vehicle_Age', 'KM_Travelled', 'Fuel_Type', 'HP',
       'Paint_Type', 'Transmission_Type', 'Engine_Size', 'Doors',
'Weight',
       'Age_Group'],
      dtype='object')
Index(['Car_Price', 'KM_Travelled', 'Fuel_Type', 'HP', 'Paint_Type',
       'Transmission_Type', 'Engine_Size', 'Doors', 'Weight',
       'Price_Category'],
      dtype='object')
```

Check Sum of NaN Values in Rows

```
dataset1.isnull().sum()
```

```
Car_Price            0
Vehicle_Age          0
KM_Travelled         0
Fuel_Type            0
HP                   0
Paint_Type           0
Transmission_Type    0
Engine_Size          0
Doors                0
Weight               0
Age_Group            0
dtype: int64
```

```
dataset2.isnull().sum()
```

```
Car_Price            0
Vehicle_Age          0
KM_Travelled         0
Fuel_Type            0
HP                   0
Paint_Type           0
Transmission_Type    0
Engine_Size          0
Doors                0
Weight               0
Price_Category       0
```

```
Age_Group            0
dtype: int64
```

```
dataset3.isnull().sum()
```

```
Car_Price            0
KM_Travelled         0
Fuel_Type            0
HP                   0
Paint_Type           0
Transmission_Type    0
Engine_Size          0
Doors                0
Weight               0
Price_Category       0
dtype: int64
```

Dataset2 Cleaning

Get the Head of Dataset1 and Daataset2

```
print (dataset1.head(2),'\n')
print (dataset2.head(2))
```

```
   Car_Price  Vehicle_Age  KM_Travelled Fuel_Type  HP Paint_Type  \
0      13500           23         46986    Diesel  90   Metallic
1      13750           23         72937    Diesel  90   Metallic

  Transmission_Type  Engine_Size  Doors  Weight Age_Group
0            Manual         2000  ثلاثة  1165.0       Old
1            Manual         2000  ثلاثة  1165.0       Old

   Car_Price  Vehicle_Age  KM_Travelled  Fuel_Type   HP Paint_Type   \
0      10500           54         63135          1  110       Main
1      11950           54         63123          1  110       Main

  Transmission_Type  Engine_Size  Doors   Weight Price_Category
Age_Group
0            Manual         1600  three    1050         Medium
0
1            Manual         1600   four    1035         Medium
0
```

Fuel_Type Column

```
dataset1.Fuel_Type.unique()
```

```
array(['Diesel', 'Petrol', 'CNG'], dtype=object)
```

```
dataset2.Fuel_Type.unique()
```

```
array([1, 0, 2], dtype=int64)
```

```
for i in range (0,len(dataset1)):
    if dataset1.loc[i,'Fuel_Type']=='Diesel':
        dataset1.loc[i,'Fuel_Type']=0
    elif dataset1.loc[i,'Fuel_Type']=='Petrol':
        dataset1.loc[i,'Fuel_Type']=1
    elif dataset1.loc[i,'Fuel_Type']=='CNG':
        dataset1.loc[i,'Fuel_Type']=2

dataset1['Fuel_Type'] = dataset1['Fuel_Type'].astype('int64')
dataset1.Fuel_Type.unique()
```

```
array([0, 1, 2], dtype=int64)
```

Paint_Type Column

```
dataset1.Paint_Type.unique()
```

```
array(['Metallic', 'Non-Metallic'], dtype=object)
```

```
dataset2.Paint_Type.unique()
```

```
array(['Main', 'Alternative'], dtype=object)
```

```
for i in range (0,len(dataset2)):
    if dataset2.loc[i,'Paint_Type']=='Main':
        dataset2.loc[i,'Paint_Type']='Metallic'
    elif dataset2.loc[i,'Paint_Type']=='Alternative':
        dataset2.loc[i,'Paint_Type']='Non-Metallic'

dataset2.Paint_Type.unique()
```

```
array(['Metallic', 'Non-Metallic'], dtype=object)
```

Doors Column

```
dataset1.Doors.unique()
```

```
array(['ثلاثة', 'خمسة', 'أربعة'], dtype=object)
```

```
dataset2.Doors.unique()
```

```
array(['three', 'four', 'five', 'two'], dtype=object)
```

```
for i in range (0,len(dataset1)):
    if dataset1.loc[i,'Doors']=='ثلاثة':
        dataset1.loc[i,'Doors']=3
    elif dataset1.loc[i,'Doors']=='أربعة':
        dataset1.loc[i,'Doors']=4
    elif dataset1.loc[i,'Doors']=='خمسة':
        dataset1.loc[i,'Doors']=5
```

```
for i in range (0,len(dataset2)):
    if dataset2.loc[i,'Doors']=='three':
        dataset2.loc[i,'Doors']=3
    elif dataset2.loc[i,'Doors']=='four':
        dataset2.loc[i,'Doors']=4
    elif dataset2.loc[i,'Doors']=='five':
        dataset2.loc[i,'Doors']=5
    elif dataset2.loc[i,'Doors']=='two':
        dataset2.loc[i,'Doors']=2

dataset1['Doors']=dataset1['Doors'].astype('int64')
dataset2['Doors']=dataset2['Doors'].astype('int64')

dataset1.Doors.unique()
```

```
array([3, 5, 4], dtype=int64)
```

```
dataset2.Doors.unique()
```

```
array([3, 4, 5, 2], dtype=int64)
```

Weight Column

```
dataset1['Weight'].dtype
```

```
dtype('float64')
```

```
dataset2['Weight'].dtype
```

```
dtype('int64')
```

```
dataset1.Weight.unique()
```

```
array([1165., 1170., 1245., 1185., 1105., 1065., 1120., 1100., 1255.,
       1270., 1110., 1195., 1180., 1075., 1130., 1275., 1060., 1115.,
       1265., 1260., 1125., 1155., 1045., 1480., 1320., 1280., 1135.,
       1090., 1150., 1085., 1160., 1205., 1084., 1140., 1095., 1025.,
       1119., 1080., 1121., 1615., 1067., 1040., 1030., 1055., 1050.,
       1103., 1070., 1035., 1015.])
```

```
dataset1['Weight'] = dataset1['Weight'].astype('int64')
dataset1['Weight'].dtype
```

```
dtype('int64')
```

```
dataset1.Weight.unique()
```

```
array([1165, 1170, 1245, 1185, 1105, 1065, 1120, 1100, 1255, 1270,
1110,
       1195, 1180, 1075, 1130, 1275, 1060, 1115, 1265, 1260, 1125,
1155,
       1045, 1480, 1320, 1280, 1135, 1090, 1150, 1085, 1160, 1205,
1084,
       1140, 1095, 1025, 1119, 1080, 1121, 1615, 1067, 1040, 1030,
1055,
       1050, 1103, 1070, 1035, 1015], dtype=int64)
```

Age Group Column

```
dataset1.Age_Group.unique()
```

```
array(['Old', 'New', 'Moderate'], dtype=object)
```

```
dataset2.Age_Group.unique()
```

```
array([0], dtype=int64)
```

```
import numpy as np
def generate_random_score(category):
   if category == 'New':
       return np.random.randint(1, 6)
   elif category == 'Moderate':
       return np.random.randint(6, 11)
   elif category == 'Old':
       return np.random.randint(11, 100)
   else:
       return np.nan


dataset1['Age_Group'] =
dataset1['Age_Group'].apply(generate_random_score)
```

```
dataset1['Age_Group'] = dataset1['Age_Group'].astype('int64')
```

```
dataset1.Age_Group.unique()
```

```
array([14, 92, 96, 21, 66, 27, 64, 51, 20, 78, 38, 97, 57, 39, 93, 76,
13,
       37, 73, 61, 36, 35, 16, 65, 47, 91, 33, 59, 84, 68, 69, 77, 62,
95,
       34, 26, 31, 29, 55, 25, 46, 42, 60, 86, 52, 90, 32, 41, 50, 80,
71,
       98, 79, 75, 87, 70, 99, 89,  2,  4, 10,  8,  9,  6, 28, 24, 88,
44,
       63, 30, 58, 23, 82, 19, 17, 83,  7,  3, 12, 45, 94, 54, 15, 85,
49,
       56, 67, 11, 48, 81, 53, 18, 74, 72, 40, 43, 22], dtype=int64)
```

Dataset3 Cleaning

Get the Head of Datasets

```
dataset1.head(2)
```

```
   Car_Price  Vehicle_Age  KM_Travelled  Fuel_Type  HP Paint_Type  \
0      13500           23         46986          0  90   Metallic   
1      13750           23         72937          0  90   Metallic   

  Transmission_Type  Engine_Size  Doors  Weight  Age_Group  
0            Manual         2000      3    1165         14  
1            Manual         2000      3    1165         92  
```

```
dataset2.head(2)
```

```
   Car_Price  Vehicle_Age  KM_Travelled  Fuel_Type   HP Paint_Type  \
0      10500           54         63135          1  110   Metallic   
1      11950           54         63123          1  110   Metallic   

  Transmission_Type  Engine_Size  Doors  Weight Price_Category 
Age_Group  
0            Manual         1600      3    1050         Medium 
0  
1            Manual         1600      4    1035         Medium 
0  
```

```
dataset3.head(2)
```

```
   Car_Price  KM_Travelled  Fuel_Type   HP Paint_Type 
Transmission_Type  \
0      10950         51421          1  110  Secondary 
Auto   
1       8950         51235          1   86    Primary 
Manual   

   Engine_Size  Doors  Weight Price_Category  
0         1600      5    1105         Medium  
1         1300      4    1000            Low  
```

Paint_Type Columns

```
dataset3.Paint_Type.unique()
```

```
array(['Secondary', 'Primary'], dtype=object)
```

```
dataset1.Paint_Type.unique()
```

```
array(['Metallic', 'Non-Metallic'], dtype=object)
```

```
for i in range (0,len(dataset3)):
    if dataset3.loc[i,'Paint_Type']=='Secondary':
        dataset3.loc[i,'Paint_Type']='Non-Metallic'
    elif dataset3.loc[i,'Paint_Type']=='Primary':
        dataset3.loc[i,'Paint_Type']='Metallic'

dataset3.Paint_Type.unique()

array(['Non-Metallic', 'Metallic'], dtype=object)
```

Get Shape of Datasets

```
dataset1.shape

(478, 11)

dataset2.shape

(478, 12)

dataset3.shape

(480, 10)
```

Drop Columns from Datasets

```
dataset1.drop(columns = ['Age_Group'] ,inplace = True)
dataset2.drop(columns = ['Age_Group','Price_Category'] ,inplace =
True)
dataset3.drop(columns = ['Price_Category'] ,inplace = True)

dataset1.shape

(478, 10)

dataset2.shape

(478, 10)

dataset3.shape

(480, 9)
```

Get Head From Datasets

```
dataset1.head(2)

   Car_Price  Vehicle_Age  KM_Travelled  Fuel_Type  HP Paint_Type  \
0      13500           23         46986          0  90   Metallic
1      13750           23         72937          0  90   Metallic

  Transmission_Type  Engine_Size  Doors  Weight
```

```
0            Manual         2000      3    1165
1            Manual         2000      3    1165
```

```
dataset2.head(2)
```

```
   Car_Price  Vehicle_Age  KM_Travelled  Fuel_Type   HP Paint_Type  \
0      10500           54         63135          1  110   Metallic
1      11950           54         63123          1  110   Metallic

  Transmission_Type  Engine_Size  Doors  Weight
0            Manual         1600      3    1050
1            Manual         1600      4    1035
```

```
dataset3.head(3)
```

```
   Car_Price  KM_Travelled  Fuel_Type   HP    Paint_Type
Transmission_Type  \
0      10950         51421          1  110  Non-Metallic
Auto
1       8950         51235          1   86      Metallic
Manual
2       8950         51000          1   86      Metallic
Manual

   Engine_Size  Doors  Weight
0         1600      5    1105
1         1300      4    1000
2         1300      3    1015
```

Know Information After Cleaning Datasets

```
dataset1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 478 entries, 0 to 477
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Car_Price          478 non-null    int64
 1   Vehicle_Age        478 non-null    int64
 2   KM_Travelled       478 non-null    int64
 3   Fuel_Type          478 non-null    int64
 4   HP                 478 non-null    int64
 5   Paint_Type         478 non-null    object
 6   Transmission_Type  478 non-null    object
 7   Engine_Size        478 non-null    int64
 8   Doors              478 non-null    int64
 9   Weight             478 non-null    int64
```

```
dtypes: int64(8), object(2)
memory usage: 37.5+ KB
```

```
dataset2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 478 entries, 0 to 477
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Car_Price          478 non-null    int64 
 1   Vehicle_Age        478 non-null    int64 
 2   KM_Travelled       478 non-null    int64 
 3   Fuel_Type          478 non-null    int64 
 4   HP                 478 non-null    int64 
 5   Paint_Type         478 non-null    object
 6   Transmission_Type  478 non-null    object
 7   Engine_Size        478 non-null    int64 
 8   Doors              478 non-null    int64 
 9   Weight             478 non-null    int64 
dtypes: int64(8), object(2)
memory usage: 37.5+ KB
```

```
dataset3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 480 entries, 0 to 479
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Car_Price          480 non-null    int64 
 1   KM_Travelled       480 non-null    int64 
 2   Fuel_Type          480 non-null    int64 
 3   HP                 480 non-null    int64 
 4   Paint_Type         480 non-null    object
 5   Transmission_Type  480 non-null    object
 6   Engine_Size        480 non-null    int64 
 7   Doors              480 non-null    int64 
 8   Weight             480 non-null    int64 
dtypes: int64(7), object(2)
memory usage: 33.9+ KB
```

Integrate My Datasets

```
newDataSet = pd.concat([dataset1,dataset2,dataset3],ignore_index = 
True)

dataset1.shape
```

```
(478, 10)

dataset2.shape

(478, 10)

dataset3.shape

(480, 9)

newDataSet.shape

(1436, 10)
```

Check my New Dataset

```
newDataSet

      Car_Price  Vehicle_Age  KM_Travelled  Fuel_Type   HP
Paint_Type  \
0         13500         23.0         46986          0   90
Metallic
1         13750         23.0         72937          0   90
Metallic
2         13950         24.0         41711          0   90
Metallic
3         14950         26.0         48000          0   90  Non-
Metallic
4         13750         30.0         38500          0   90  Non-
Metallic
...         ...          ...           ...        ...  ...           .
..
1431       7500          NaN         20544          1   86
Metallic
1432      10845          NaN         11000          1   86  Non-
Metallic
1433       8500          NaN         17016          1   86  Non-
Metallic
1434       7250          NaN         11000          1   86
Metallic
1435       6950          NaN             1          1  110  Non-
Metallic

     Transmission_Type  Engine_Size  Doors  Weight
0               Manual         2000      3    1165
1               Manual         2000      3    1165
2               Manual         2000      3    1165
3               Manual         2000      3    1165
4               Manual         2000      3    1170
```

```
...                ...          ...     ...      ...
1431            Manual         1300       3     1025
1432            Manual         1300       3     1015
1433            Manual         1300       3     1015
1434            Manual         1300       3     1015
1435            Manual         1600       5     1114

[1436 rows x 10 columns]
```

```
newDataSet.isnull().sum()
```

```
Car_Price              0
Vehicle_Age          480
KM_Travelled           0
Fuel_Type              0
HP                     0
Paint_Type             0
Transmission_Type      0
Engine_Size            0
Doors                  0
Weight                 0
dtype: int64
```

Clean my New Dataset

Put the mean insted of NaN Values

```
newDataSet['Vehicle_Age']=newDataSet['Vehicle_Age'].fillna(value=newDataSet['Vehicle_Age'].mean())

newDataSet['Vehicle_Age'] = newDataSet['Vehicle_Age'].astype('int64')


newDataSet
```

```
      Car_Price  Vehicle_Age  KM_Travelled  Fuel_Type   HP  
Paint_Type  \
0         13500           23         46986          0   90  
Metallic
1         13750           23         72937          0   90  
Metallic
2         13950           24         41711          0   90  
Metallic
3         14950           26         48000          0   90   Non-
Metallic
4         13750           30         38500          0   90   Non-
Metallic
...         ...          ...           ...        ...  ...            .
```

```
..
1431       7500          47        20544         1   86  
Metallic
1432      10845          47        11000         1   86  Non-
Metallic
1433       8500          47        17016         1   86  Non-
Metallic
1434       7250          47        11000         1   86  
Metallic
1435       6950          47            1         1  110  Non-
Metallic

     Transmission_Type  Engine_Size  Doors  Weight
0               Manual         2000      3    1165
1               Manual         2000      3    1165
2               Manual         2000      3    1165
3               Manual         2000      3    1165
4               Manual         2000      3    1170
...                ...          ...    ...     ...
1431            Manual         1300      3    1025
1432            Manual         1300      3    1015
1433            Manual         1300      3    1015
1434            Manual         1300      3    1015
1435            Manual         1600      5    1114

[1436 rows x 10 columns]
```

Check My New Dataset

```
newDataSet.isnull().sum()
```

```
Car_Price            0
Vehicle_Age          0
KM_Travelled         0
Fuel_Type            0
HP                   0
Paint_Type           0
Transmission_Type    0
Engine_Size          0
Doors                0
Weight               0
dtype: int64
```

```
newDataSet.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1436 entries, 0 to 1435
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
```

```
 0   Car_Price          1436 non-null   int64 
 1   Vehicle_Age        1436 non-null   int64 
 2   KM_Travelled       1436 non-null   int64 
 3   Fuel_Type          1436 non-null   int64 
 4   HP                 1436 non-null   int64 
 5   Paint_Type         1436 non-null   object
 6   Transmission_Type  1436 non-null   object
 7   Engine_Size        1436 non-null   int64 
 8   Doors              1436 non-null   int64 
 9   Weight             1436 non-null   int64 
dtypes: int64(8), object(2)
memory usage: 112.3+ KB
```

Write To New DataSets

```
newDataSet.to_csv('dataintegration.csv')

print ("Save Successfuly")
```

```
Save Successfuly
```