Faculty Of Computers and Artificial Intelligence

Cairo University

212202.FCI.AI496.Selected Topics in Artificial intelligence-2

Assignment (3)

AYA SABRY MOHAMED

2018035

Submitted to
Eng.Salah Mostafa

May 2022

Colab links

https://colab.research.google.com/drive/1UyhF-HWr7C5NzdCLgutuyIxAdEdFIVAe?usp=sharing

(finished with jupyter code and pdf)

- df.head(5)

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino

- df1.info()

```
class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64
 2   displacement  398 non-null    float64
 3   horsepower    398 non-null    int64
 4   weight        398 non-null    int64
 5   acceleration  398 non-null    float64
 6   model year    398 non-null    int64
 7   origin        398 non-null    int64
 8   car name      398 non-null    object
 9   Type          398 non-null    int8
dtypes: float64(3), int64(5), int8(1), object(1)
memory usage: 28.5+ KB
```

- *Use regular expression to extract only 2-3 first characters from column car name*🚗

```
df1['Type'] = df['car name'].str.extract('(.{2,3})', expand=False)
print(df1['Type'].T.value_counts())
```

```
1  df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
mpg	398.0	23.514573	7.815984	9.0	17.500	23.0	29.000	46.6
cylinders	398.0	5.454774	1.701004	3.0	4.000	4.0	8.000	8.0
displacement	398.0	193.425879	104.269838	68.0	104.250	148.5	262.000	455.0
weight	398.0	2970.424623	846.841774	1613.0	2223.750	2803.5	3608.000	5140.0
acceleration	398.0	15.568090	2.757689	8.0	13.825	15.5	17.175	24.8
model year	398.0	76.010050	3.697627	70.0	73.000	76.0	79.000	82.0
origin	398.0	1.572864	0.802055	1.0	1.000	1.0	2.000	3.0

- *Use Pearson correlation which represent MIC as in our book 📘 page(60)*

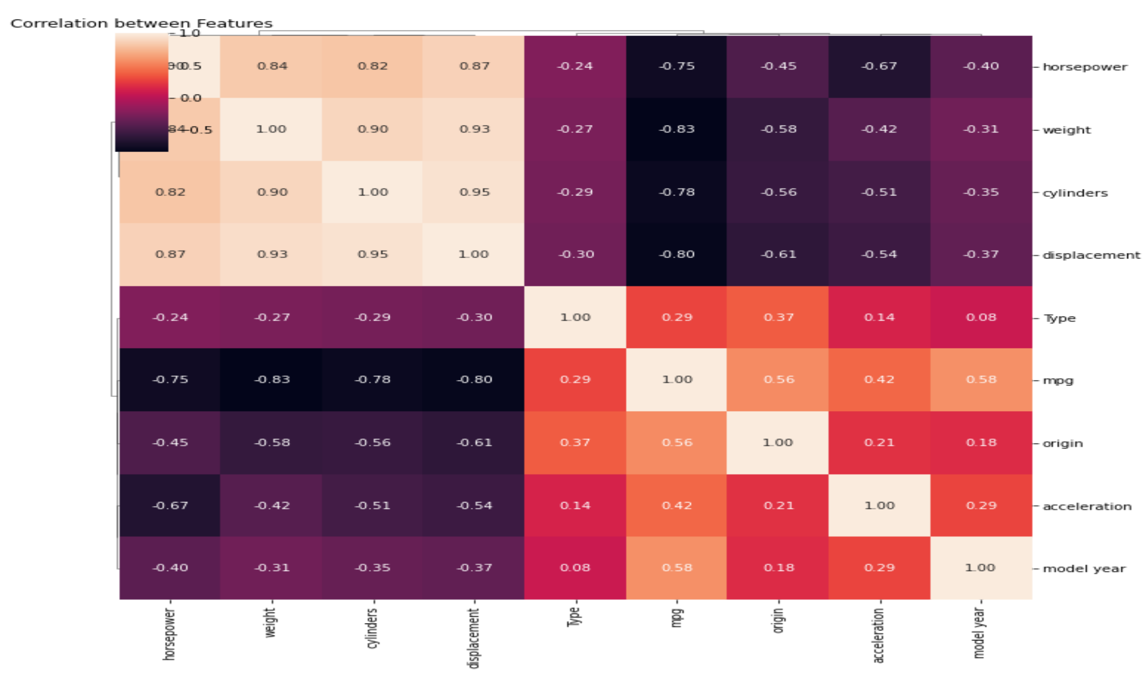
```
df1.corr(method ='pearson')
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	Type
mpg	1.000000	-0.775396	-0.804203	-0.753177	-0.831741	0.420289	0.579267	0.563450	0.288368
cylinders	-0.775396	1.000000	0.950721	0.818454	0.896017	-0.505419	-0.348746	-0.562543	-0.286512
displacement	-0.804203	0.950721	1.000000	0.873330	0.932824	-0.543684	-0.370164	-0.609409	-0.302291
horsepower	-0.753177	0.818454	0.873330	1.000000	0.841770	-0.665833	-0.397772	-0.454271	-0.236643
weight	-0.831741	0.896017	0.932824	0.841770	1.000000	-0.417457	-0.306564	-0.581024	-0.265872
acceleration	0.420289	-0.505419	-0.543684	-0.665833	-0.417457	1.000000	0.288137	0.205873	0.138012
model year	0.579267	-0.348746	-0.370164	-0.397772	-0.306564	0.288137	1.000000	0.180662	0.077134
origin	0.563450	-0.562543	-0.609409	-0.454271	-0.581024	0.205873	0.180662	1.000000	0.374745
Type	0.288368	-0.286512	-0.302291	-0.236643	-0.265872	0.138012	0.077134	0.374745	1.000000

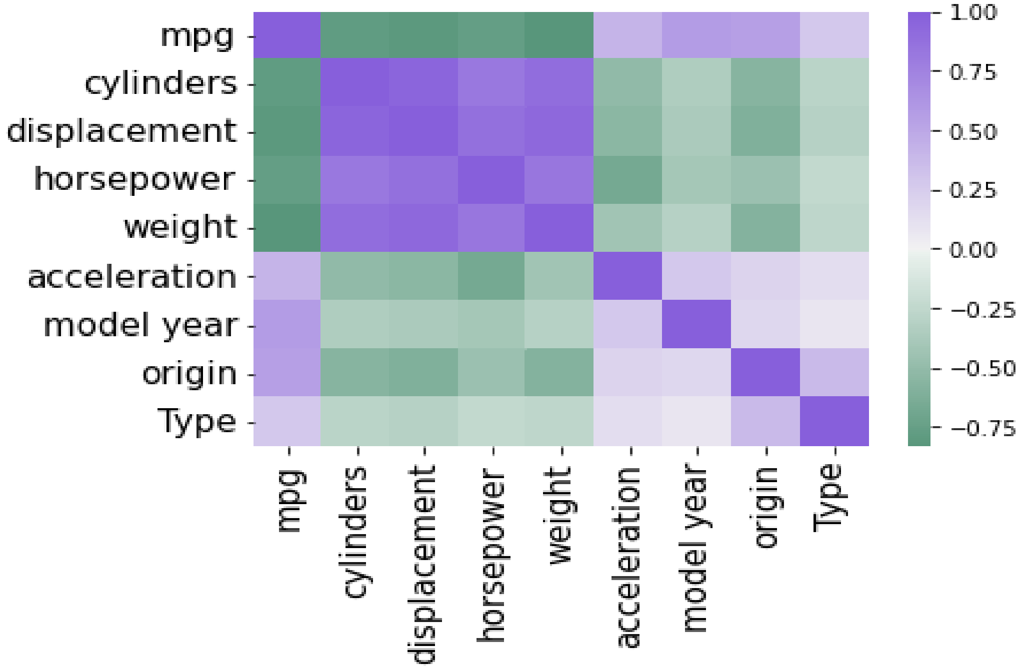
```
corr_matrix = df1.corr(method='pearson')

plt.figure(figsize=(25,25))
sns.clustermap(corr_matrix, annot=True, fmt = ".2f", dendrogram_ratio=0.01)

plt.title("Correlation between Features")
plt.show()
```



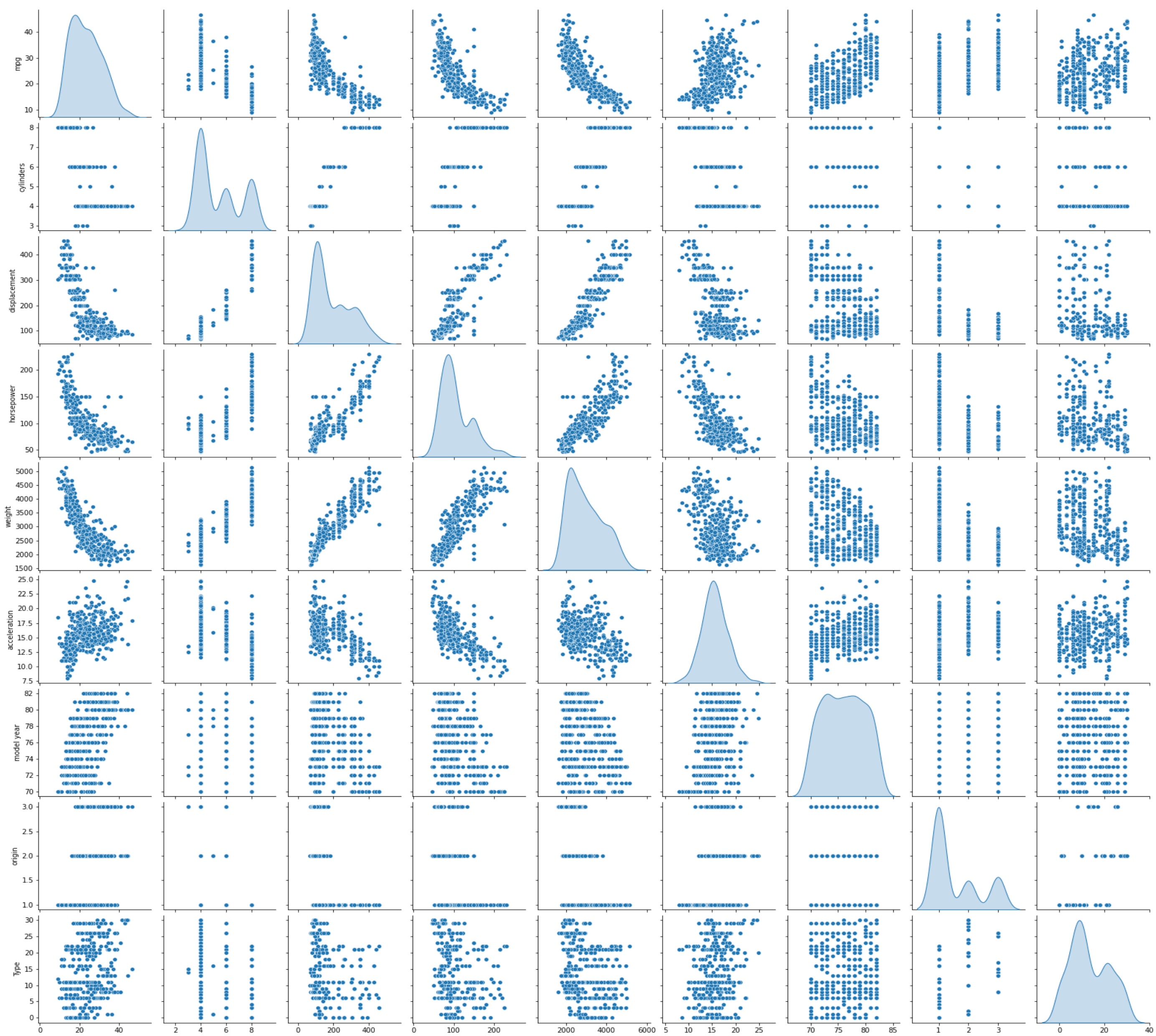
```
colors = sns.diverging_palette(150, 275, as_cmap=True)
sns.heatmap(df1.corr(), center=0, cmap=colors)
plt.xticks(fontsize= 15)
plt.yticks(fontsize= 15)
```



- **Plot allpairwise relationship** (*like book p.60*)

```
sns.pairplot(df1, diag_kind="kde")

plt.show()
```



Mutual information

- This is as Assignment was required
- From sklearn.feature_selection.mutual_info_regression
- The main advantage in this method that we can use continuous data with discrete data without any need to transformation but I use transformation just once when y= displacement (just for try).
- The term “discrete features” is used instead of naming them “categorical”, because it describes the essence more accurately. For example, pixel intensities of an image are discrete features (but hardly categorical) and you will get better results if mark them as such. Also note, that treating a continuous variable as discrete and vice versa will usually give incorrect results, so be attentive about that.
- True mutual information can’t be negative. If its estimate turns out to be negative, it is replaced by zero.

This concept based on reading from :

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_regression.html

- $I(X ; Y) = H(X) - H(X \mid Y)$

1-Mutual information between x=data and y= mpg

displacement	0.7834541975190548
weight	0.7752353889847736
horsepower	0.7173123149839054
cylinders	0.5604999092038978
model year	0.2948140934041654
origin	0.24267984255442943
Type	0.21358315745584155
acceleration	0.1326139750195825

2-Mutual information between x=data and y= displacement

displacement	1.6402015660596843
weight	0.9357700710469548
cylinders	0.8232087633850549
horsepower	0.7304796993104103
mpg	0.6357091481272019
Type	0.35972082899951596
origin	0.32673863209376197
acceleration	0.20017386087715394
model year	0.08556060457598846

3-Mutual information between x=data and y= cylinders

displacement	0.831562209829865
weight	0.6932003088647258
horsepower	0.6217627501276608
mpg	0.530709921545065
origin	0.2683126937944862
acceleration	0.22605598042718977
Type	0.2027509161573695
model year	0.1270373808088241

4-Mutual information between x=data and y= weight

displacement	0.9403041251692796
horsepower	0.7973322375414331
mpg	0.7921481147324081
cylinders	0.6880967959844455
origin	0.26056710847805964
model year	0.1655033209732637
Type	0.16078897503566658
acceleration	0.145596692307842

5-Mutual information between x=data and y= horsepower

weight	0.800373902571315
displacement	0.758967491159447
mpg	0.739396590471398
cylinders	0.685742920337184
Type	0.424295633428911
acceleration	0.325042702451782
origin	0.319165676897847
model year	0.223093224843235

6-Mutual information between x=data and y= acceleration

horsepower	0.309073611798592
cylinders	0.222890519948165
displacement	0.195548369859557
weight	0.168180064898154
mpg	0.146244360662560
origin	0.080984172377638
model year	0.046834956369482
Type	0.040651197113387

7-Mutual information between x=data and y=origin

index	0
Type	0.828337410882112
horsepower	0.32620246822876364
cylinders	0.3222918388774309
displacement	0.298019106108089
weight	0.263348948939756
mpg	0.181725191313350
acceleration	0.11576108545158359
model year	0.029941625573480124

8-Mutual information between x=data and y= model year

mpg	0.30910819895680940
horsepower	0.23386750865763384
weight	0.159126209306093
cylinders	0.08153790768753266
displacement	0.0769871229954728
acceleration	0.05708935843315732
origin	0.04178805873598912
Type	0.0

9-Mutual information between x=data and y= Type

origin	0.8547301573971928
horsepower	0.4132377717545732
displacement	0.33500552442209086
cylinders	0.2229849025808952
mpg	0.2158539090072149
weight	0.17744422336415733
acceleration	0.02772262281742588
model year	0.023860885026222967