

```
In [123]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score # Indecater for Data Q
from sklearn.preprocessing import StandardScaler # Calulate
```

```
In [4]: df = pd.read_csv('car_miles_per_gallon.csv')
```

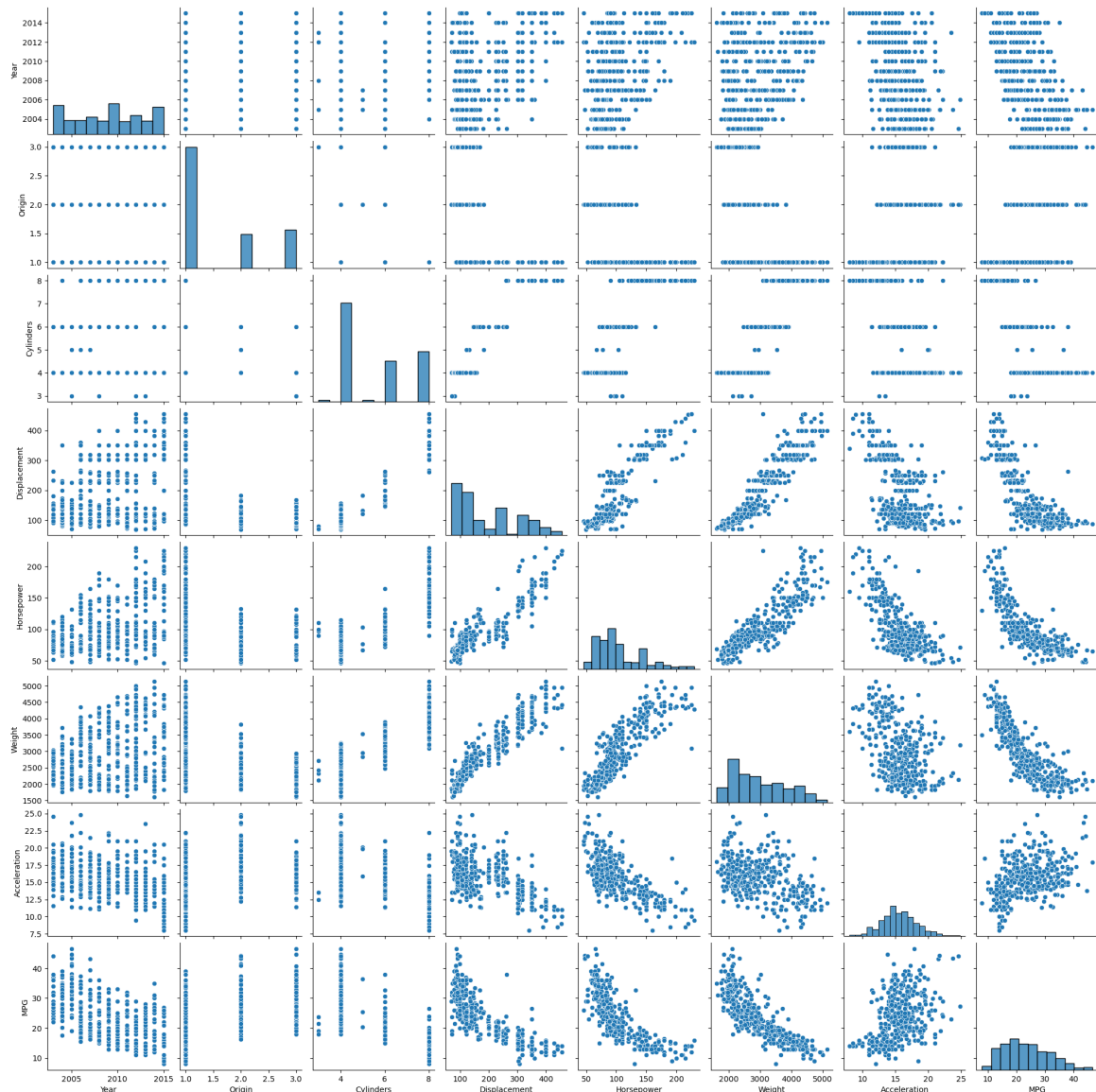
```
In [10]: df.head()
```

```
Out[10]:
```

	Name	Year	Origin	Cylinders	Displacement	Horsepower	Weight	A
0	Chevrolet Chevelle Malibu	2015	1	8	307.0	130.0	3504	
1	Buick Skylark 320	2015	1	8	350.0	165.0	3693	
2	Plymouth Satellite	2015	1	8	318.0	150.0	3436	
3	Amc Rebel Sst	2015	1	8	304.0	150.0	3433	
4	Ford Torino	2015	1	8	302.0	140.0	3449	

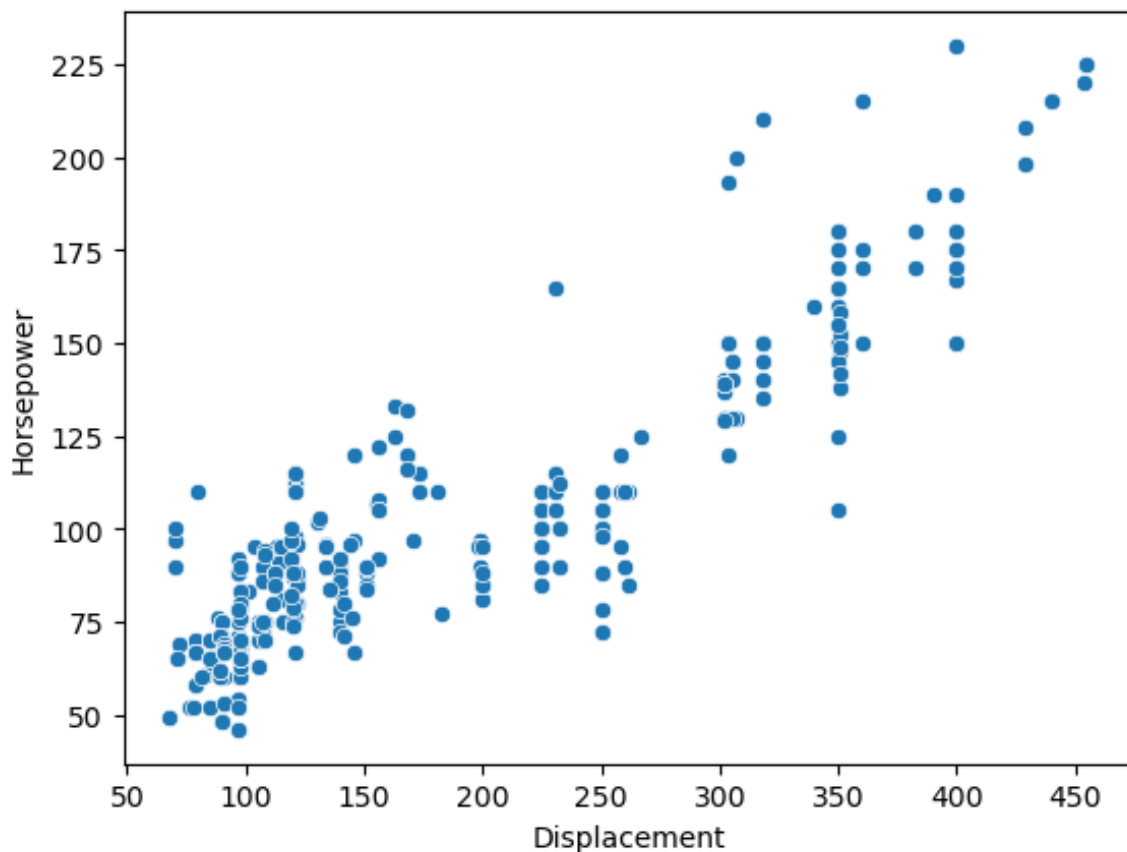
```
In [12]: sns.pairplot(df)
```

```
Out[12]: <seaborn.axisgrid.PairGrid at 0x1118f8bf0>
```



```
In [24]: sns.scatterplot(data = df, x = 'Displacement', y = 'Horsepower')
```

```
Out[24]: <Axes: xlabel='Displacement', ylabel='Horsepower'>
```



```
In [18]: df2 = df[['Displacement', 'Horsepower']].dropna()
df2.head()
```

```
Out[18]:
```

	Displacement	Horsepower
0	307.0	130.0
1	350.0	165.0
2	318.0	150.0
3	304.0	150.0
4	302.0	140.0

```
In [31]: model = KMeans(n_clusters = 4, random_state = 0) #Random start is s
model.fit(df2) #.Fit is Bring Data To Process
```

```
Out[31]:
```

▼ **KMeans** ⓘ ?

KMeans(n_clusters=4, random_state=0)

```
In [34]: model.cluster_centers_
```

```
Out[34]: array([[110.28672986, 79.50236967],
                [326.73611111, 150.05555556],
                [411.61538462, 190.61538462],
                [225.44578313, 101.40963855]])
```

```
In [42]: model.labels_
```

```

Out[42]: array([1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 0, 3, 3, 3, 0, 0,
0, 0,
           0, 0, 3, 2, 1, 1, 1, 0, 0, 0, 3, 3, 3, 3, 3, 1, 2, 1, 1, 2,
2, 2,
           3, 0, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1,
1, 1,
           2, 1, 1, 2, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
1, 1,
           1, 2, 2, 1, 1, 2, 2, 1, 3, 3, 3, 3, 3, 0, 2, 2, 1, 1, 3, 0,
0, 0,
           0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 1, 0, 0, 1, 3, 3, 3, 0, 0, 0,
0, 3,
           3, 3, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 3,
3, 3,
           2, 1, 1, 1, 3, 3, 3, 3, 3, 3, 1, 0, 0, 3, 0, 0, 0, 0, 3, 0,
3, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 3, 3, 3, 3, 0, 0, 0,
0, 3,
           3, 3, 3, 0, 0, 0, 0, 0, 1, 0, 0, 3, 1, 1, 1, 1, 0, 0, 0, 0,
0, 1,
           3, 1, 1, 3, 3, 3, 3, 2, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
           0, 0, 0, 0, 0, 3, 1, 1, 3, 3, 3, 0, 3, 3, 3, 3, 3, 3, 1, 3,
1, 1,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 3, 0, 0, 3, 3, 0, 3, 3, 1,
1, 1,
           1, 1, 1, 3, 1, 0, 0, 0, 0, 3, 1, 0, 3, 0, 0, 0, 0, 0, 3, 3,
0, 0,
           0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
           3, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
           0, 0, 0, 0, 3, 0, 3, 1, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
           0, 0, 0, 0, 0, 0, 3, 3, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0])
dtype=int32

```

```

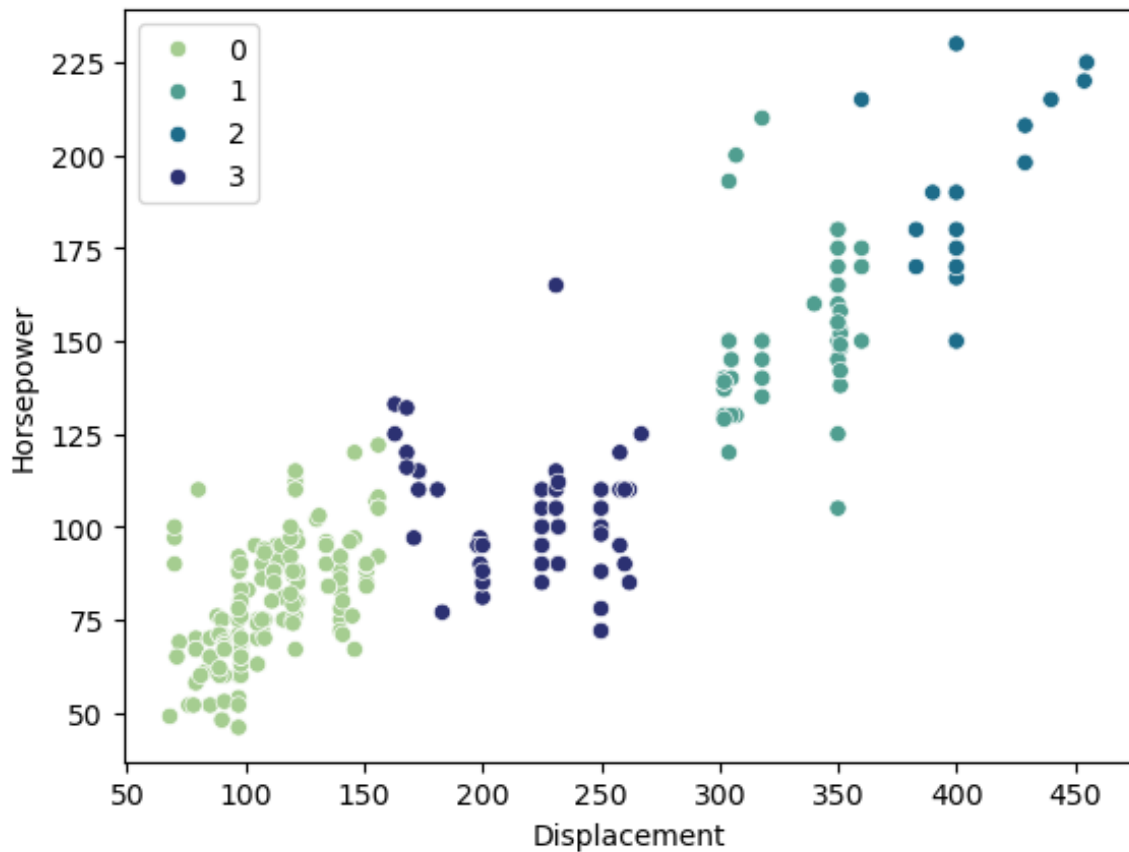
In [67]: sns.scatterplot(data = df2, x = 'Displacement', y = 'Horsepower', h

```

```

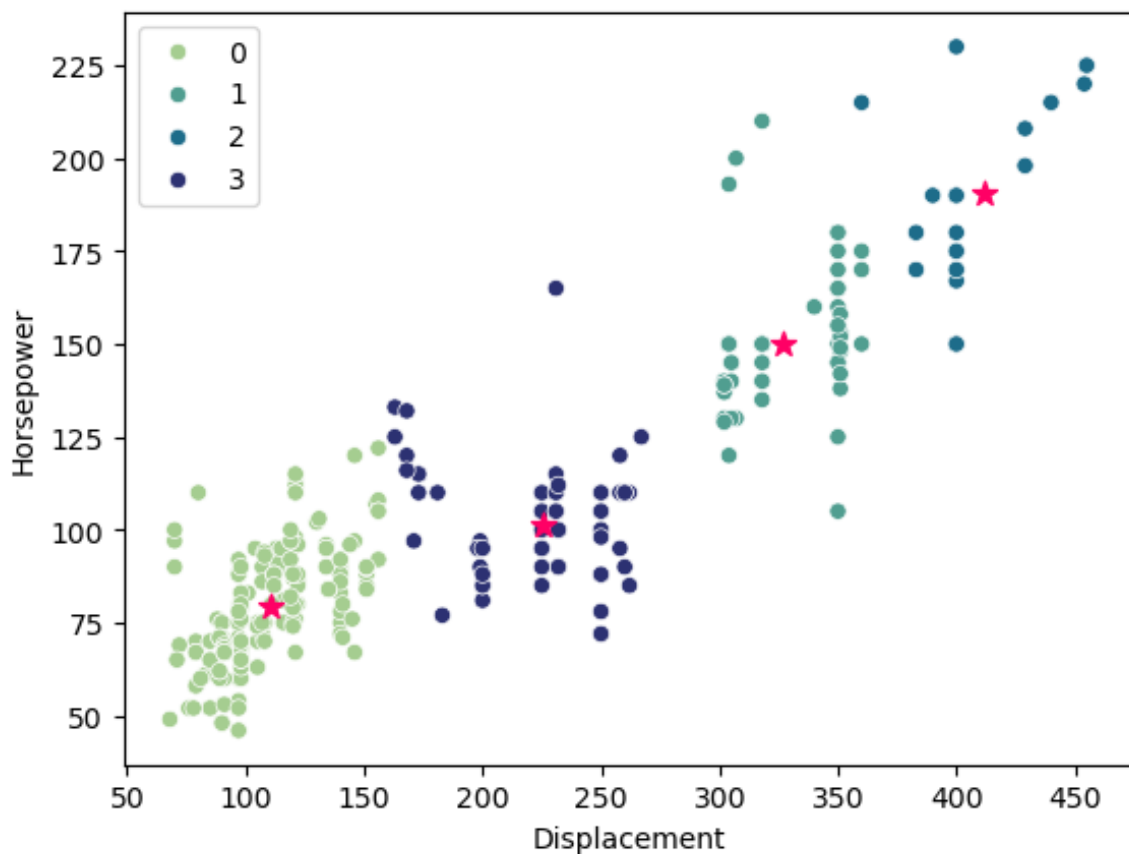
Out[67]: <Axes: xlabel='Displacement', ylabel='Horsepower'>

```



```
In [75]: sns.scatterplot(data = df2, x = 'Displacement', y = 'Horsepower', h
plt.scatter( model.cluster_centers_[0],model.cluster_centers_[1]
```

```
Out[75]: <matplotlib.collections.PathCollection at 0x15eb78530>
```



```
In [81]: score = silhouette_score(df2,model.labels_)
```

score

Out[81]: 0.6460117066143266

```
In [91]: #For Loop
df2 = df[['Displacement', 'Horsepower']].dropna()

for k in range(2,11) :
    model = KMeans(n_clusters = k, random_state = 0) #Random start
    model.fit(df2)
    score = silhouette_score(df2,model.labels_)
    print('k = {}, Score {}'.format(k,score))
```

```
k = 2, Score 0.6725041105122287
k = 3, Score 0.647577531375378
k = 4, Score 0.6460117066143266
k = 5, Score 0.6162919020799671
k = 6, Score 0.5287882758929692
k = 7, Score 0.546077412954501
k = 8, Score 0.5025214229491167
k = 9, Score 0.47644493359313916
k = 10, Score 0.46910551129143246
```

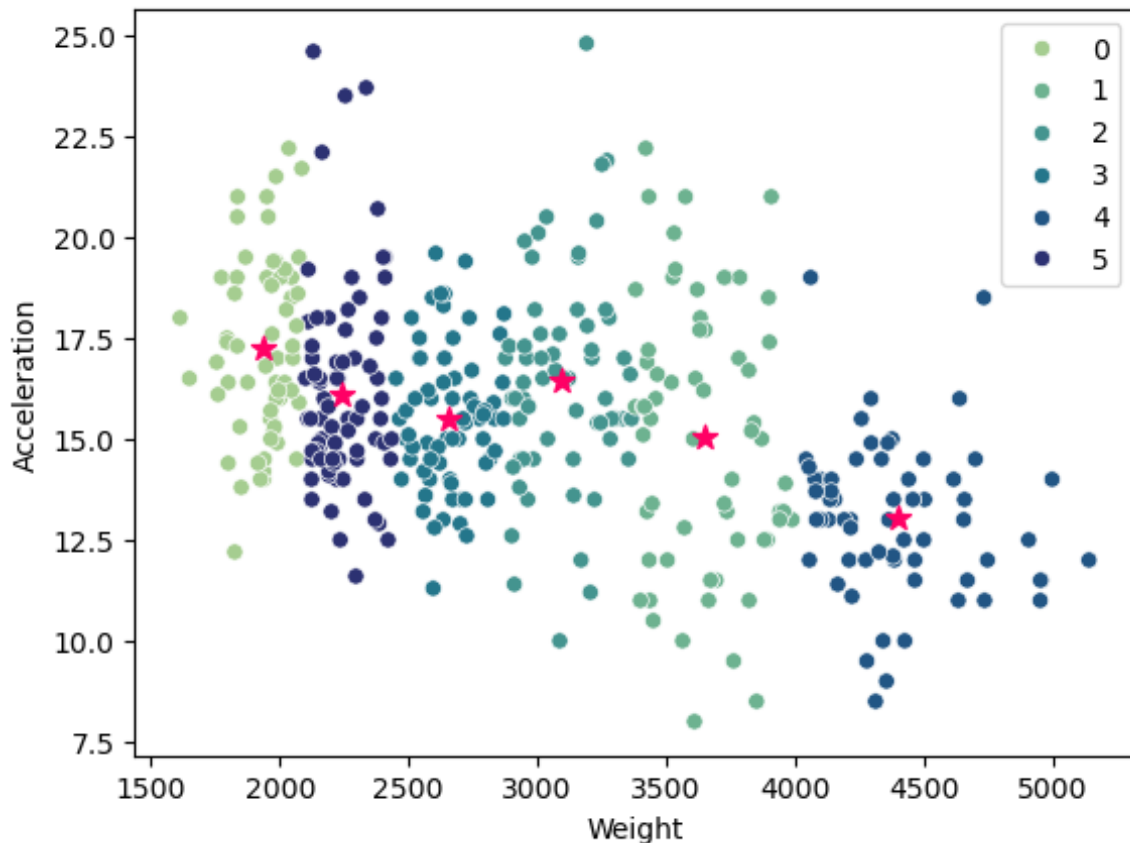
```
In [151... #For Loop
df2 = df[['Weight', 'Acceleration']].dropna()

model = KMeans(n_clusters = 6, random_state = 0) #Random start is s
model.fit(df2)

sns.scatterplot(data = df2, x = 'Weight', y = 'Acceleration', hue =
plt.scatter( model.cluster_centers_[0],model.cluster_centers_[1]

score = silhouette_score(df2,model.labels_)
print('k = {}, Score {}'.format(k,score))
```

```
k = 10, Score 0.5406605137513921
```



p1 : x = 1800, y = 12

p2 : x = 3200, y = 25

$\sqrt{(x1-x2)^2 + (y1-y2)^2}$

```
In [153... ## p1 : x = 1800, y = 12
## p2 : x = 3200, y = 25
## sqrt((x1-x2)**2 + (y1-y2)**2)
```

```
In [155... scaler = StandardScaler()
scaler.fit_transform(df2)
```

```
Out[155... array([[ 0.63086987, -1.29549834],
[ 0.85433297, -1.47703779],
[ 0.55047045, -1.65857724],
[ 0.54692342, -1.29549834],
[ 0.56584093, -1.84011669],
[ 1.62049216, -2.02165614],
[ 1.63586264, -2.38473503],
[ 1.58620417, -2.56627448],
[ 1.71980909, -2.02165614],
[ 1.03996105, -2.56627448],
[ 0.7006282 , -2.02165614],
[ 0.75501604, -2.74781393],
```

```
[ 0.93473239, -2.20319559],
[ 0.1366499 , -2.02165614],
[-0.70754402, -0.20626165],
[-0.16248324, -0.02472221],
[-0.23224156, -0.02472221],
[-0.45333997,  0.15681724],
[-0.99367137, -0.3878011 ],
[-1.34246298,  1.79067228],
[-0.35284069,  0.70143559],
[-0.63896804, -0.3878011 ],
[-0.70399699,  0.70143559],
[-0.87070755, -1.1139589 ],
[-0.38121696, -0.20626165],
[ 1.94445453, -0.56934055],
[ 1.66187422, -0.20626165],
[ 1.66896828, -0.75088   ],
[ 2.08278883,  1.06451448],
[-0.99367137, -0.3878011 ],
[-0.83523722, -0.02472221],
[-0.87780162, -0.56934055],
[-1.0929883 ,  1.24605393],
[-0.39776978, -0.93241945],
[ 0.55401749, -0.02472221],
[ 0.4239596 , -0.02472221],
[ 0.3920363 , -0.02472221],
[ 0.37548348, -0.02472221],
[ 1.4644227 , -1.29549834],
[ 1.76592053, -1.47703779],
[ 1.39939375, -0.75088   ],
[ 1.33081778, -0.93241945],
[ 2.34645164, -1.47703779],
[ 2.09934165, -1.29549834],
[ 2.56518536, -1.29549834],
[-0.00996081, -0.75088   ],
[-0.66497962,  1.24605393],
[ 0.36838941, -0.20626165],
[ 0.19931416, -0.3878011 ],
[-0.88726037, -0.56934055],
[-1.00194778, -0.56934055],
[-1.05988266,  1.42759338],
[-1.07052376, -0.3878011 ],
[-1.41576833,  1.24605393],
[-1.60494344,  0.88297504],
[-1.34364532,  1.24605393],
[-1.20058165,  1.79067228],
[-0.8186844 , -0.02472221],
[-0.99840075,  0.51989614],
[-0.84706066,  2.87990897],
[-0.66497962,  1.42759338],
[-0.88016631,  0.33835669],
[ 1.54127508, -1.29549834],
[ 1.67251532, -1.29549834],
[ 1.37692921, -0.75088   ],
[ 1.36983514, -0.93241945],
[ 0.82950374, -1.47703779],
[ 1.96573673, -1.65857724],
```



```
[ 1.81084961, -0.75088 ],
[ 1.75646177, -0.75088 ],
[ 1.71626206, -1.1139589 ],
[-0.75720249, -0.75088 ],
[ 1.08961951, -1.1139589 ],
[ 1.33318246, -0.56934055],
[ 1.56492197, 0.15681724],
[ 1.30835323, -0.56934055],
[-0.04424879, -0.3878011 ],
[-0.54319814, 0.88297504],
[ 0.01013905, 1.42759338],
[-0.92391305, 0.88297504],
[-0.6803501 , 0.15681724],
[-0.80686095, 0.51989614],
[-0.54910987, -0.3878011 ],
[-0.95347166, -0.20626165],
[-1.0291417 , 0.33835669],
[ 1.33554715, -0.93241945],
[ 0.82950374, -1.47703779],
[ 1.20312458, -0.93241945],
[ 1.26697118, -0.3878011 ],
[ 0.9536499 , -1.1139589 ],
[ 2.34290461, -1.47703779],
[ 1.76592053, -1.29549834],
[ 1.64650374, -0.93241945],
[ 1.49752834, -0.3878011 ],
[ 2.08633587, -1.65857724],
[ 2.34172226, -1.65857724],
[ 1.00567306, -1.65857724],
[ 0.17803196, 0.33835669],
[ 0.36366003, 0.88297504],
[-0.03006066, 0.15681724],
[ 0.05979752, 0.33835669],
[-0.07853678, 0.15681724],
[-1.20649337, 1.97221172],
[ 2.39611011, -0.56934055],
[ 2.28851676, -1.1139589 ],
[ 1.99056597, -0.93241945],
[ 1.80730258, -1.1139589 ],
[-0.21450639, -0.20626165],
[-0.81750205, 1.24605393],
[-0.67325603, 1.42759338],
[-0.69926761, 0.33835669],
[-1.00076544, -0.75088 ],
[-0.78084937, 1.06451448],
[-0.58930958, -0.56934055],
[-0.83405487, -0.02472221],
[ 1.31426495, -0.93241945],
[ 1.54600446, -2.20319559],
[-1.30462796, 1.42759338],
[-0.96056573, -0.02472221],
[-0.45925169, -0.56934055],
[-0.12110118, -0.02472221],
[ 0.50672371, -1.65857724],
[-0.36702882, -0.56934055],
[-0.19322419, -0.75088 ],
```

```
[ 0.82004498, -1.65857724],  
[ 0.15556741,  0.33835669],  
[-0.11282477,  0.51989614],  
[-0.08208382,  0.15681724],  
[ 0.43223601,  0.51989614],  
[-1.20649337,  1.24605393],  
[-0.61413881,  0.33835669],  
[-1.34128063,  1.97221172],  
[-0.50654547,  0.51989614],  
[ 0.95837928,  0.51989614],  
[ 0.78220996,  0.88297504],  
[ 0.75974542,  0.33835669],  
[ 1.38402328, -0.56934055],  
[ 2.04377147, -0.3878011 ],  
[ 1.75764411, -0.75088   ],  
[ 1.97164846,  0.15681724],  
[ 1.52117523, -0.02472221],  
[-0.88844272,  0.33835669],  
[-1.19112289, -0.02472221],  
[-0.79267282, -0.3878011 ],  
[-1.56237904,  0.33835669],  
[-1.14382911,  1.24605393],  
[-0.99958309, -0.3878011 ],  
[-1.01968295, -0.02472221],  
[-0.85651942, -0.56934055],  
[-0.56920972, -0.20626165],  
[-0.68507948, -0.02472221],  
[-1.14737615,  0.15681724],  
[ 0.34710721,  0.15681724],  
[ 0.57766437,  0.15681724],  
[ 0.54574108,  1.97221172],  
[ 0.2217787 ,  1.42759338],  
[ 2.00711879, -1.47703779],  
[ 1.73754426, -0.56934055],  
[ 1.80612024, -0.3878011 ],  
[ 1.994113 , -0.75088   ],  
[ 1.10735468,  1.97221172],  
[ 1.09553123,  1.06451448],  
[ 0.89807972,  1.24605393],  
[ 0.96310866,  1.24605393],  
[ 0.08107971, -0.20626165],  
[ 0.2962664 , -0.75088   ],  
[ 0.23478449, -1.29549834],  
[-0.94519525,  0.15681724],  
[-0.39185806,  0.51989614],  
[-0.06671334,  0.15681724],  
[-0.44742824,  1.06451448],  
[-0.31737036, -0.75088   ],  
[-0.88371334,  0.33835669],  
[-0.50299843,  0.51989614],  
[ 0.01605077, -0.3878011 ],  
[-1.22186385, -0.56934055],  
[ 0.28444296,  0.51989614],  
[-0.32682911, -0.20626165],  
[-0.01587253,  0.51989614],  
[-0.03006066, -0.3878011 ],
```

[-0.35402303, -0.75088],
[-1.38975675, 0.70143559],
[-0.59876833, -0.02472221],
[-0.88726037, 0.48358825],
[-0.47107513, -0.24256954],
[-0.84587832, 0.77405137],
[-0.90854257, -0.09733799],
[1.47151676, -0.93241945],
[1.44195815, -0.93241945],
[1.17238362, -0.60564844],
[1.47151676, -1.00503523],
[0.31045453, -0.0610301],
[0.45233587, -0.3878011],
[0.04915642, 0.73774348],
[0.13546756, 0.73774348],
[-1.10599409, 2.4079064],
[-0.95347166, 2.37159851],
[-1.22186385, -0.49672477],
[-1.38975675, 0.6651277],
[0.80467451, 0.77405137],
[0.71363398, 1.97221172],
[0.79758044, 0.22943302],
[0.26316076, 0.81035926],
[-1.35428642, -1.22288256],
[-1.15919959, 0.51989614],
[-0.96411276, 0.3020488],
[-0.47935154, -0.71457211],
[0.21231995, 0.04789357],
[1.14637205, -0.85980367],
[0.35420128, 2.29898273],
[-0.04779583, -0.02472221],
[1.00449071, 0.41097247],
[1.66660359, -1.25919045],
[1.28234165, -1.29549834],
[1.06360794, -0.20626165],
[0.92763833, -0.56934055],
[-1.09417065, 1.06451448],
[-0.96411276, -0.27887743],
[-1.35428642, 1.10082237],
[-0.79267282, -0.02472221],
[-1.21240509, 0.44728036],
[1.07543138, -1.1139589],
[1.28825338, 1.24605393],
[1.38284093, -0.67826422],
[1.56610432, -0.24256954],
[0.64978739, 0.3020488],
[0.53746466, 0.48358825],
[0.77984527, 0.77405137],
[0.65569911, 1.24605393],
[1.47742849, -1.62226935],
[1.41239954, -1.51334568],
[1.60157465, -1.22288256],
[1.61339809, -0.3878011],
[-1.21831681, -0.3878011],
[-0.27244127, 0.15681724],
[-0.83405487, 0.95559081],

[-0.2547061 , 0.08420146],
[-1.08707658, 0.51989614],
[-1.05870031, 0.12050935],
[-1.16511131, 0.3020488],
[-0.92273071, -0.53303266],
[-0.18376544, -0.3878011],
[-0.43796949, -1.00503523],
[-0.29608816, -0.75088],
[-1.16511131, 2.15375117],
[-1.38384503, -0.42410899],
[-1.16511131, 1.39128549],
[-1.06461204, 1.10082237],
[-1.38384503, 0.3020488],
[0.466524 , -0.02472221],
[0.90399144, -0.85980367],
[0.70890461, -1.00503523],
[0.66752255, 1.31866971],
[0.21823167, 0.95559081],
[-0.00641377, 0.08420146],
[-0.29608816, -0.0610301],
[0.54337639, 0.59251192],
[0.28326061, 0.59251192],
[0.48425917, 0.08420146],
[0.11773239, 0.41097247],
[0.76802183, 1.13713026],
[0.5197295 , -0.16995376],
[0.53746466, -0.85980367],
[0.56111155, -0.78718789],
[0.27734889, -1.58596146],
[1.31190027, -0.67826422],
[-0.96411276, 0.33835669],
[-0.48526327, -0.49672477],
[-0.79267282, -0.31518532],
[-0.87543693, -0.3878011],
[-0.53846877, -0.27887743],
[-0.26652955, 0.41097247],
[-0.13647166, 0.73774348],
[-0.66852665, -0.24256954],
[-0.16603027, 0.12050935],
[0.2004965 , -0.71457211],
[-0.20741233, 0.04789357],
[0.5197295 , 0.08420146],
[-1.15919959, -0.24256954],
[-0.98775965, 0.37466458],
[0.32464267, -0.0610301],
[0.02314484, 0.95559081],
[-0.09508961, 0.62881981],
[0.34828956, 0.95559081],
[0.46061228, 0.37466458],
[1.0281376 , -0.0610301],
[0.89216799, -0.78718789],
[1.16410721, -0.85980367],
[1.01631416, -0.13364588],
[1.64295671, -0.24256954],
[1.28115931, -0.46041688],
[0.75028666, -0.20626165],

```
[ 1.14637205, -0.93241945],  
[-1.23605198, -0.56934055],  
[-1.17693476, -0.13364588],  
[-1.24787542, -0.42410899],  
[-0.35520538, -0.20626165],  
[ 0.66161083, 1.64544072],  
[ 1.09907827, 0.6651277 ],  
[ 0.25961372, 3.35191153],  
[ 0.53155294, 2.4079064 ],  
[-0.91090726, -0.85980367],  
[-0.97002448, -0.24256954],  
[-1.12372926, 1.31866971],  
[-0.99367137, -0.31518532],  
[-0.35520538, 0.15681724],  
[-0.44388121, -1.54965357],  
[-0.31973505, -0.96872734],  
[-0.48999264, -0.85980367],  
[-0.97711855, -0.31518532],  
[-1.18521117, 1.17343815],  
[-1.00549482, -0.02472221],  
[-1.1249116 , 0.3020488 ],  
[-0.34574662, 0.33835669],  
[-0.11873649, 0.91928292],  
[ 0.03851532, 1.64544072],  
[ 0.48544151, 1.13713026],  
[-0.92509539, 0.08420146],  
[-0.30672926, -0.02472221],  
[-0.50654547, 0.70143559],  
[-0.63423866, -0.20626165],  
[-0.83405487, -0.13364588],  
[-1.01731826, 0.84666715],  
[-0.2015006 , -0.42410899],  
[-1.01731826, 1.31866971],  
[-1.04687687, 2.22636695],  
[-0.75129076, 2.95252475],  
[-0.02414894, 1.57282494],  
[ 0.33055439, 2.26267484],  
[-1.32472781, -0.64195633],  
[-1.34246298, 0.62881981],  
[-0.9759362 , 0.88297504],  
[-1.33063953, -0.09733799],  
[-0.07144272, -1.51334568],  
[-0.65079149, -1.1139589 ],  
[-0.55620393, -0.16995376],  
[-0.07735444, -0.46041688],  
[-0.80449626, 0.51989614],  
[-0.56802738, 0.04789357],  
[-0.39658743, 0.3020488 ],  
[-0.4143226 , -0.42410899],  
[-0.29017644, -1.07765101],  
[-0.69217354, -0.96872734],  
[-1.43705053, 0.48358825],  
[-1.2951692 , 0.3020488 ],  
[-1.43113881, 0.19312513],  
[-1.07052376, 0.81035926],  
[-1.17693476, 1.39128549],
```

```
[ -1.08825893,  0.62881981],
[ -1.16511131,  0.15681724],
[ -0.89317209, -0.24256954],
[ -1.09417065,  0.22943302],
[ -0.69808526,  1.86328806],
[ -0.92273071, -0.49672477],
[ -0.76902593,  0.08420146],
[ -0.89908382, -0.42410899],
[ -0.7335556 ,  0.44728036],
[ -0.42023432, -0.27887743],
[ -0.39658743,  0.9918987 ],
[  0.3069075 ,  1.75436439],
[  0.22414339,  1.46390127],
[ -0.08326616, -1.07765101],
[ -0.04779583, -0.64195633],
[  0.52564122,  0.08420146],
[  0.89216799,  1.24605393],
[  0.10590895,  0.55620403],
[  0.58475844,  0.37466458],
[ -0.43205777,  1.46390127],
[ -0.39067571,  1.10082237],
[ -0.6803501 ,  0.88297504],
[ -0.4675281 ,  0.22943302],
[ -0.52664532,  0.15681724],
[ -0.27835299,  0.88297504],
[ -0.12464822,  0.3020488 ],
[  0.07635034,  1.79067228],
[ -1.17102304, -0.09733799],
[ -1.11781754,  0.95559081],
[ -1.18284648,  0.73774348],
[ -0.99958309, -0.31518532],
[ -0.99958309,  0.62881981],
[ -0.95820104, -0.3878011 ],
[ -0.90499554, -0.3878011 ],
[ -0.85770176,  0.48358825],
[ -1.1887582 , -0.20626165],
[ -1.1887582 ,  0.04789357],
[ -1.15328787,  0.22943302],
[ -0.03006066,  0.3020488 ],
[  0.05270345,  0.51989614],
[ -0.45570466, -0.3878011 ],
[ -0.16011855, -0.31518532],
[ -0.3611171 , -0.60564844],
[ -0.70990871, -0.93241945],
[ -0.02414894,  0.62881981],
[ -0.21332405,  0.01158568],
[ -0.99367137,  3.27929575],
[ -0.79858454, -1.4407299 ],
[ -0.40841088,  1.10082237],
[ -0.29608816,  1.39128549]])
```

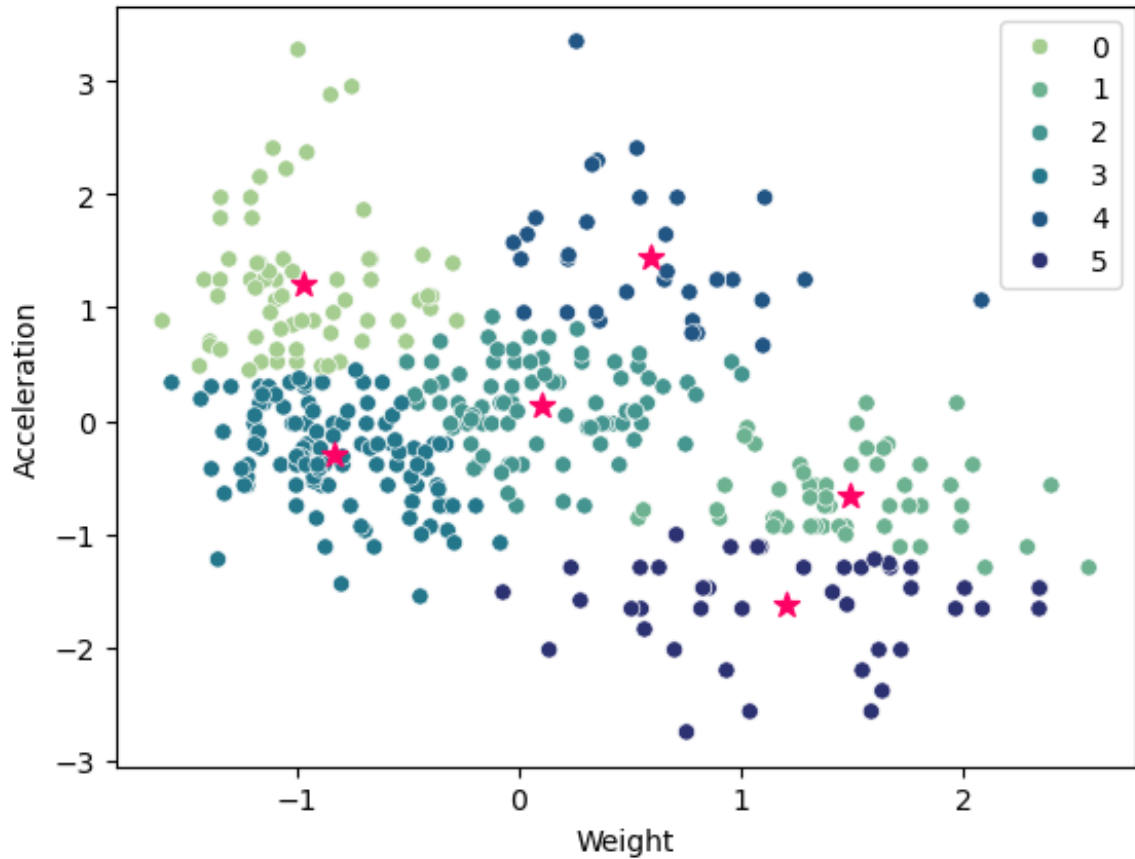
```
In [157... scaler = StandardScaler()
df2[['Weight','Acceleration']] = scaler.fit_transform(df2)
```

```
In [159... model = KMeans(n_clusters = 6, random_state = 0) #Random start is s
model.fit(df2)
```

```
sns.scatterplot(data = df2, x = 'Weight', y = 'Acceleration', hue =
plt.scatter( model.cluster_centers_[0],model.cluster_centers_[1]

score = silhouette_score(df2,model.labels_)
print('k = {}, Score {}'.format(k,score))
```

k = 10, Score 0.364486106828356



In [161... df2.head()

Out[161...

	Weight	Acceleration
0	0.630870	-1.295498
1	0.854333	-1.477038
2	0.550470	-1.658577
3	0.546923	-1.295498
4	0.565841	-1.840117

In []:

In []:

In [164... scaler.transform([[3500,20]])

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/base.py:493: Use
rWarning: X does not have valid feature names, but StandardScaler wa
s fitted with feature names
warnings.warn(
```

```
Out[164... array([[0.6261405 , 1.60913283]])
```

```
In [166... model.predict( scaler.transform( [[3500,20]] ))
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/base.py:493: Use
rWarning: X does not have valid feature names, but StandardScaler wa
s fitted with feature names
warnings.warn(
/opt/anaconda3/lib/python3.12/site-packages/sklearn/base.py:493: Use
rWarning: X does not have valid feature names, but KMeans was fitted
with feature names
warnings.warn(
```

```
Out[166... array([4], dtype=int32)
```

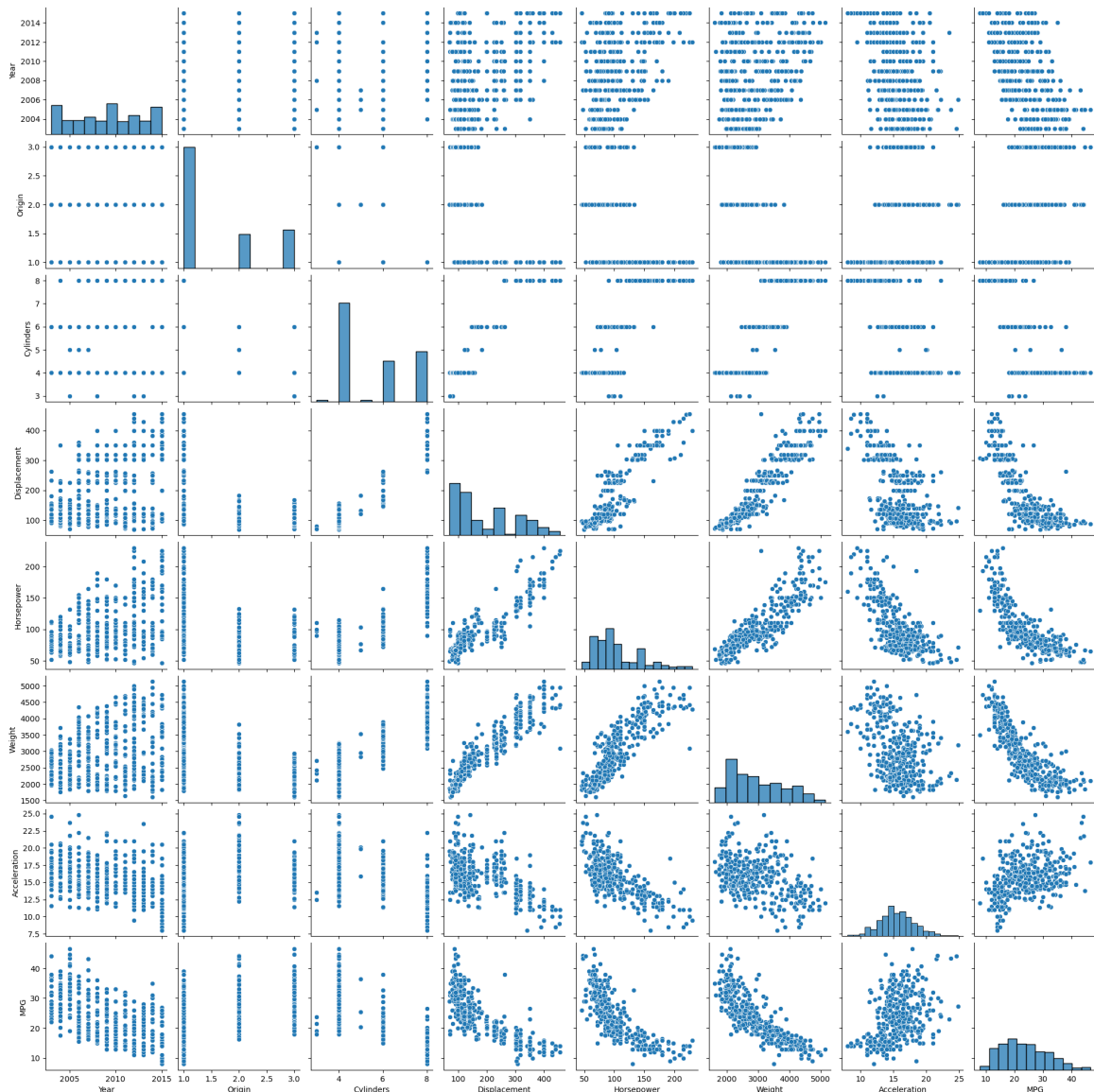
Home Work

1. สร้าง K-Mean จาก Data โดยใช้ 2 Column พร้อม Plot Graph

```
In [ ]:
```

```
In [193... sns.pairplot(df)
```

```
Out[193... <seaborn.axisgrid.PairGrid at 0x3085015e0>
```

In [256..

```

model = KMeans(n_clusters = 2, random_state = 0)
model.fit(DFH)

DFH = pd.DataFrame()

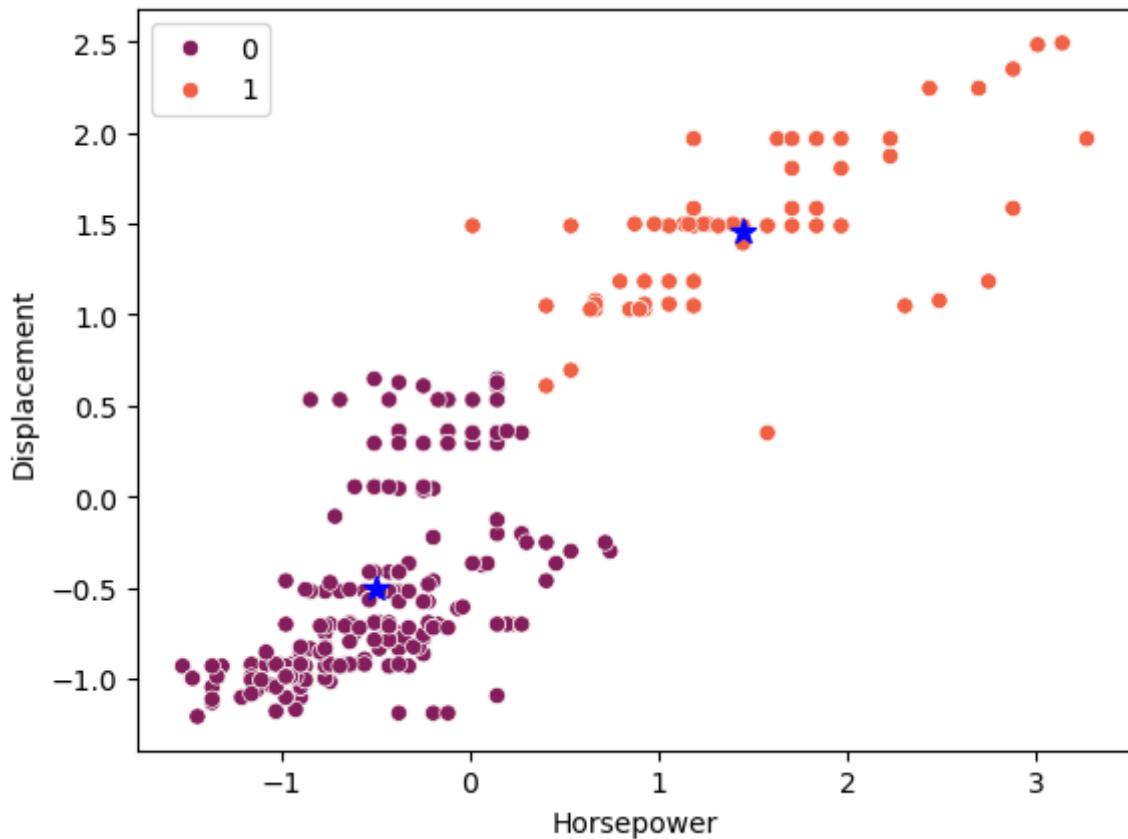
DFH = df[['Horsepower', 'Displacement']].dropna()

scaler = StandardScaler()
DFH[['Horsepower', 'Displacement']] = scaler.fit_transform(DFH)

sns.scatterplot(data = DFH, x = 'Horsepower', y = 'Displacement', h
plt.scatter( model.cluster_centers_[0], model.cluster_centers_[1]
print('Score = {}'.format(score))

```

Score = {} 0.3484308952448908



2. สร้าง K-Mean จาก Data โดยใช้ 3 Column ขึ้นไปและหาค่า Score ที่ดีที่สุด

```
In [270... CDF = df[['Displacement', 'Horsepower', 'Displacement', 'Weight']].dro

best_score = -1
best_k = None

for k in range(2, 11) :
    model = KMeans(n_clusters = k, random_state = 0) #Random start
    model.fit(df2)
    score = silhouette_score(df2, model.labels_)
    print('k = {}, Score {}'.format(k, score))

    if score > best_score :
        best_score = score
        best_k = k

print(f'\nBest Score = {best_score} at k = {best_k}')
```

k = 2, Score 0.4915568111184752
k = 3, Score 0.46607447608339053
k = 4, Score 0.39462775921107723
k = 5, Score 0.3559665405212341
k = 6, Score 0.3547524032647868
k = 7, Score 0.343811658873796
k = 8, Score 0.33111728429802584
k = 9, Score 0.3404634563486486
k = 10, Score 0.3484308952448908

Best Score = 0.4915568111184752 at k = 2

In []: