تمرین دوم داده کاوی

۲-۱:

دانلود شد.

۲-۲:

In [3]:
```
import numpy as np
import pandas as pd
```

In [4]:
```
df = pd.read_csv('dataset_54_vehicle.csv', sep=',')
df.head()
```

Out[4]:

	COMPACTNESS	CIRCULARITY	DISTANCE_CIRCULARITY	RADIUS_RATIO	PR.AXIS_ASPECT_RATIO	MAX.LENGTH_ASPECT_RATIO	SCATTER_RATIO	ELONG/
0	95	48	83	178	72	10	162	
1	91	41	84	141	57	9	149	
2	104	50	106	209	66	10	207	
3	93	41	82	159	63	9	144	
4	85	44	70	205	103	52	149	

۲-۳:

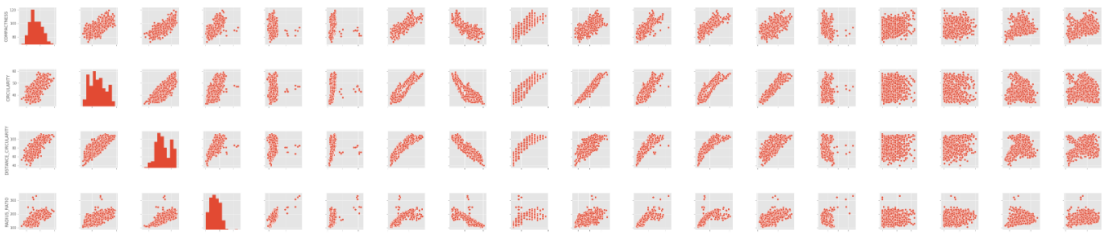
In [5]:
```
attribute = 'Class'
variables = df[attribute].unique()    #This gives different features in that attribute (like 'Sweet')
print (variables)
```

```
['van' 'saab' 'bus' 'opel']
```

۲-۴:

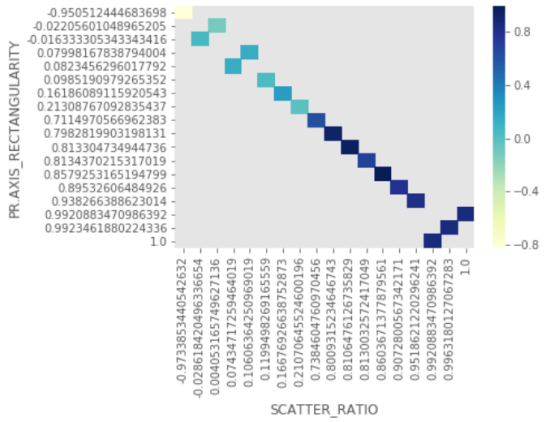
In [6]:
```
import numpy as np
import pandas as pd
from sklearn import preprocessing
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
matplotlib.style.use('ggplot')

df = pd.read_csv('dataset_54_vehicle.csv', sep=',')
df.head(10)
sns.pairplot(df)
```

Out[6]:
```
<seaborn.axisgrid.PairGrid at 0x1e170a80da0>
```



In [9]:
```
df.head()
heatmap1_data = pd.pivot_table(df.corr(), values='CIRCULARITY', index=['PR.AXIS_RECTANGULARITY'], columns='SCATTER_RATI
sns.heatmap(heatmap1_data, cmap="YlGnBu")
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e10392d048>
```



In [7]:
```
df.corr(method ='pearson').style.format("{:.2}").background_gradient(cmap=plt.get_cmap('coolwarm'), axis=1)
```

Out[7]:

	COMPACTNESS	CIRCULARITY	DISTANCE_CIRCULARITY	RADIUS_RATIO	PR.AXIS_ASPECT_RATIO	MAX.LENGTH_ASPECT_RA
COMPACTNESS	1.0	0.69	0.79	0.69	0.093	
CIRCULARITY	0.69	1.0	0.8	0.62	0.15	
DISTANCE_CIRCULARITY	0.79	0.8	1.0	0.77	0.16	
RADIUS_RATIO	0.69	0.62	0.77	1.0	0.67	
PR.AXIS_ASPECT_RATIO	0.093	0.15	0.16	0.67	1.0	
MAX.LENGTH_ASPECT_RATIO	0.15	0.25	0.26	0.45	0.65	
SCATTER_RATIO	0.81	0.86	0.91	0.74	0.11	
ELONGATEDNESS	-0.79	-0.83	-0.91	-0.79	-0.19	
PR.AXIS_RECTANGULARITY	0.81	0.86	0.9	0.71	0.08	
MAX.LENGTH_RECTANGULARITY	0.68	0.97	0.77	0.57	0.13	
SCALED_VARIANCE_MAJOR	0.76	0.81	0.86	0.8	0.27	
SCALED_VARIANCE_MINOR	0.82	0.85	0.89	0.73	0.092	
SCALED_RADIUS_OF_GYRATION	0.59	0.94	0.71	0.54	0.12	
SKEWNESS_ABOUT_MAJOR	-0.25	0.059	-0.23	-0.18	0.15	
SKEWNESS_ABOUT_MINOR	0.23	0.15	0.12	0.051	-0.057	0.
KURTOSIS_ABOUT_MAJOR	0.16	-0.015	0.26	0.17	-0.034	0.
KURTOSIS_ABOUT_MINOR	0.3	-0.11	0.15	0.38	0.24	-0.
HOLLOWS_RATIO	0.37	0.039	0.34	0.47	0.27	

۲-۵ ---- ۲-۶: با هم انجام شده است:

In [5]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split

df = pd.read_csv('dataset_54_vehicle.csv', sep=',')
#print(df.head().T)
print(df.shape)
data = df.iloc[:,:-1]
label = df.iloc[:,18]
data_train, data_test, labels_train, labels_test = train_test_split(data, label, test_size=0.2, random_state=42)
print(data_train.shape)
print(data_test.shape)
```

```
(846, 19)
(676, 18)
(170, 18)
```

۲-۷:

In [2]:
```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree


model = tree.DecisionTreeClassifier(criterion='entropy', max_depth=5,
            max_features=4)
model
```

Out[2]:
```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=5,
            max_features=4, max_leaf_nodes=None, min_impurity_decrease=0.0,
            min_impurity_split=None, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            presort=False, random_state=None, splitter='best')
```

۲-۸:

In [29]:
```
import random
from scipy.stats import randint
from sklearn.tree import DecisionTreeClassifier
params = {"max_depth": [3, None],
              "max_features": randint(1, 9),
              "min_samples_leaf": randint(1, 9)}
params
##model = DecisionTreeClassifier(**params)
```

Out[29]:
```
{'max_depth': [3, None],
 'max_features': <scipy.stats._distn_infrastructure.rv_frozen at 0x1e103a5bc18>,
 'min_samples_leaf': <scipy.stats._distn_infrastructure.rv_frozen at 0x1e103a5bf98>}
```

۲-۹ : ----- ۲-۱۰: با هم انجام شده است:

In [30]:
```
from scipy.stats import randint
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import RandomizedSearchCV



# Instantiate a Decision Tree classifier: tree
tree = DecisionTreeClassifier()

# Instantiate the RandomizedSearchCV object: tree_cv
tree_cv = RandomizedSearchCV(tree, params, cv=5)

# Fit it to the data
tree_cv.fit(data, label)

# Print the tuned parameters and score
print("Tuned Decision Tree Parameters: {}".format(tree_cv.best_params_))
print("Best score is {}".format(tree_cv.best_score_))
```

```
Tuned Decision Tree Parameters: {'max_depth': None, 'max_features': 5, 'min_samples_leaf': 3}
Best score is 0.6832151300236406
```

۲-۱۲:

هر چه cv بالا برود مدل بهتر آموزش میبیند.

۲-۱۳:

In [32]:
```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree
model = tree.DecisionTreeClassifier( max_depth = None, max_features= 5,min_samples_leaf= 3)
model.fit(data, label)
model.predict(data_test)
```

Out[32]:
```
array(['bus', 'van', 'bus', 'saab', 'bus', 'van', 'van', 'saab', 'bus',
       'opel', 'saab', 'bus', 'saab', 'opel', 'opel', 'van', 'saab',
       'bus', 'saab', 'opel', 'van', 'van', 'saab', 'opel', 'opel',
       'saab', 'opel', 'van', 'van', 'van', 'saab', 'van', 'bus', 'van',
       'van', 'bus', 'bus', 'saab', 'bus', 'saab', 'van', 'bus', 'saab',
       'van', 'saab', 'opel', 'bus', 'van', 'bus', 'van', 'bus', 'bus',
       'bus', 'opel', 'bus', 'opel', 'opel', 'opel', 'saab', 'saab',
       'bus', 'van', 'saab', 'bus', 'bus', 'bus', 'saab', 'saab', 'opel',
       'opel', 'saab', 'bus', 'saab', 'bus', 'van', 'van', 'saab', 'opel',
       'saab', 'van', 'bus', 'van', 'saab', 'opel', 'saab', 'opel',
       'opel', 'bus', 'van', 'saab', 'saab', 'opel', 'bus', 'bus', 'opel',
       'saab', 'van', 'van', 'bus', 'opel', 'saab', 'saab', 'saab', 'van',
       'bus', 'bus', 'saab', 'van', 'saab', 'opel', 'van', 'bus', 'van',
       'bus', 'bus', 'van', 'bus', 'opel', 'bus', 'opel', 'bus', 'opel',
       'saab', 'van', 'bus', 'saab', 'bus', 'van', 'bus', 'opel', 'bus',
       'van', 'bus', 'saab', 'bus', 'bus', 'bus', 'saab', 'saab', 'bus',
       'van', 'saab', 'bus', 'bus', 'van', 'opel', 'saab', 'opel', 'saab',
       'opel', 'saab', 'bus', 'bus', 'saab', 'van', 'van', 'bus', 'van',
       'opel', 'bus', 'bus', 'van', 'opel', 'saab', 'opel', 'opel',
       'saab', 'opel', 'saab', 'van', 'van', 'opel', 'saab', 'saab',
       'bus', 'bus', 'opel', 'saab', 'van', 'van', 'bus', 'bus', 'bus',
       'saab', 'bus', 'van', 'saab', 'van', 'bus', 'bus', 'van', 'bus',
       'saab', 'saab', 'van', 'opel', 'bus', 'van', 'opel', 'saab',
       'saab', 'van', 'bus', 'bus', 'bus', 'opel', 'bus', 'opel', 'saab',
       'bus', 'saab', 'opel'], dtype=object)
```

feature_importances_

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

Out[34]:
```
array([0.03165779, 0.02389382, 0.07200121, 0.028007  , 0.02706205,
       0.16645216, 0.04389276, 0.24751246, 0.01107652, 0.05836371,
       0.07140115, 0.02817715, 0.02180019, 0.0481998 , 0.04630108,
       0.03218704, 0.01636789, 0.02564621])
```

۲-۱۴: ---- ۲-۱۵ : ------ ۲:۱۶ : باهم انجام شده است:

In [14]:
```
from sklearn.tree import export_graphviz
# Export as dot file
export_graphviz(model,
                out_file='dot_data.dot',
                feature_names = data.columns,
                class_names = 'Class',
                rounded = True, proportion = False,
                precision = 2, filled = True)
```

In [15]:
```
try:
    from StringIO import StringIO
except ImportError:
    from io import StringIO
from sklearn import tree
import pydotplus

dotfile = StringIO()
tree.export_graphviz(model,
                out_file=dotfile,
                feature_names = data.columns,
                class_names = 'Class',
                rounded = True, proportion = False,
                precision = 2, filled = True)
graph=pydotplus.graph_from_dot_data(dotfile.getvalue())
graph.write_png("dtree.png")
```

Out[15]: True

In [16]:
```
graph.write_pdf("dtree.pdf")
```

Out[16]: True