In [11]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import scipy.stats as sts
import matplotlib as mpl

%matplotlib inline
```

In [14]:
```
from sklearn.decomposition import FactorAnalysis as fact
from sklearn.decomposition import PCA as pca
```

In [16]:
```
#Clustering modules
import sklearn.metrics as metcs
from scipy.cluster import hierarchy as hier
from sklearn import cluster as cls
```

In [87]:
```
#For the tree
from sklearn.feature_extraction.image import grid_to_graph
from sklearn import tree
from sklearn.externals.six import StringIO
from IPython.display import Image
import pydotplus
```

In [5]:
```
path = "C:/Users/Administrator/Documents/Master/MSIS-5223-70250 - Programming
 for Data Sci - 8282017 - 159 PM/Data for Tutorials and ICE/Data"
os.chdir(path)
df = pd.read_table('car.test.frame.txt', sep= '\t')
```

In [27]:
```
df = df.dropna()
```

In [28]:
```
df1 = df.copy()
```

In [29]:
```
df1.columns
```

Out[29]:
```
Index(['Price', 'Country', 'Reliability', 'Mileage', 'Type', 'Weight', 'Dis
p.',
       'HP'],
      dtype='object')
```

In [89]:
```
df1.dtypes
```

Out[89]:
```
Price            int64
Country         object
Reliability    float64
Mileage          int64
Type          category
Weight           int64
Disp.            int64
HP               int64
dtype: object
```

In [90]:
```
df1.head()
```

Out[90]:

	Price	Country	Reliability	Mileage	Type	Weight	Disp.	HP
0	8895	USA	4.0	33	Small	2560	97	113
1	7402	USA	2.0	33	Small	2345	114	90
2	6319	Korea	4.0	37	Small	1845	81	63
3	6635	Japan/USA	5.0	32	Small	2260	91	92
4	6599	Japan	5.0	32	Small	2440	113	103

In [91]:
```
rows, cols = df1.shape
rows, cols
```

Out[91]: (49, 8)

In [43]:
```
df1.head()
```

Out[43]:

	Price	Country	Reliability	Mileage	Type	Weight	Disp.	HP
0	8895	USA	4.0	33	Small	2560	97	113
1	7402	USA	2.0	33	Small	2345	114	90
2	6319	Korea	4.0	37	Small	1845	81	63
3	6635	Japan/USA	5.0	32	Small	2260	91	92
4	6599	Japan	5.0	32	Small	2440	113	103

In [49]:
```
df1.describe()
```

Out[49]:

	Price	Reliability	Mileage	Weight	Disp.	HP
count	49.000000	49.000000	49.000000	49.000000	49.000000	49.000000
mean	12452.081633	3.387755	24.795918	2897.755102	156.714286	125.408163
std	4229.656447	1.455111	4.813088	500.257289	57.374138	32.514688
min	6319.000000	1.000000	18.000000	1845.000000	81.000000	63.000000
25%	9599.000000	2.000000	21.000000	2560.000000	114.000000	102.000000
50%	11650.000000	3.000000	23.000000	2885.000000	146.000000	115.000000
75%	14944.000000	5.000000	27.000000	3265.000000	181.000000	150.000000
max	24760.000000	5.000000	37.000000	3855.000000	305.000000	225.000000

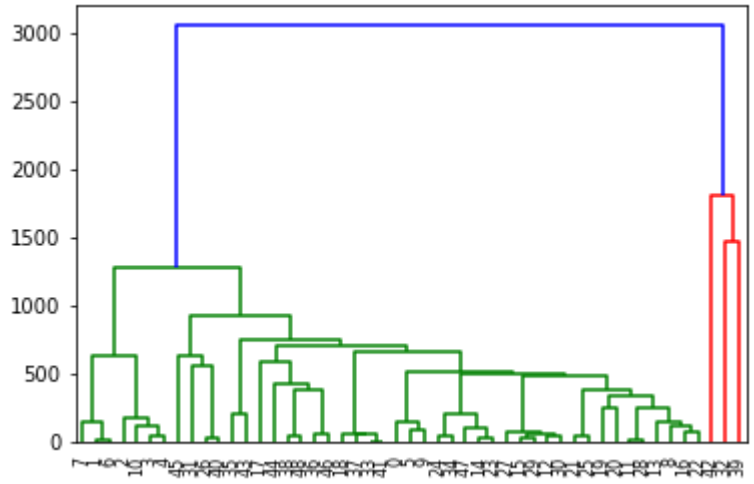
In [52]:
```
#Use 6 clusters
km = cls.KMeans(n_clusters=6).fit(df1.loc[:,['Price','Mileage']])
km.labels_      #assigned clusters
```

Out[52]:
```
array([1, 5, 5, 5, 5, 1, 5, 5, 1, 1, 5, 1, 3, 1, 3, 3, 1, 0, 3, 1, 1, 3, 1,
       3, 3, 3, 4, 3, 1, 3, 3, 4, 2, 3, 3, 0, 0, 3, 0, 2, 4, 3, 2, 0, 0, 4,
       0, 3, 0])
```

In [54]:
```
#Use 4 clusters
km = cls.KMeans(n_clusters=4).fit(df1.loc[:,['Price','Mileage']])
km.labels_      #assigned clusters
```

Out[54]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 2, 0, 2, 2, 2, 2, 2, 0,
       2, 2, 2, 1, 2, 0, 2, 2, 1, 3, 2, 2, 1, 1, 2, 1, 3, 1, 2, 3, 1, 1, 1,
       1, 2, 1])
```

In [58]:
```
#Create a plot to view the output
z = hier.linkage(df1[['Price','Mileage']], 'single')
plt.figure()
dn = hier.dendrogram(z)
```



In [106]:
```
df1['Type'] = df1['Type'] .astype('category')
```

In [105]:
```
col_names = list(df1.columns.values)
classnames = list(df1.Type.unique())
```

Out[105]:

	Price	Country	Reliability	Mileage	Type	Weight	Disp.	HP
0	8895	USA	4.0	33	Small	2560	97	113
1	7402	USA	2.0	33	Small	2345	114	90
2	6319	Korea	4.0	37	Small	1845	81	63
3	6635	Japan/USA	5.0	32	Small	2260	91	92
4	6599	Japan	5.0	32	Small	2440	113	103
5	8672	Mexico	4.0	26	Small	2285	97	82
6	7399	Japan/USA	5.0	33	Small	2275	97	90
7	7254	Korea	1.0	28	Small	2350	98	74
8	9599	Japan	5.0	25	Small	2295	109	90
10	8748	Japan/USA	5.0	29	Small	2390	97	102
11	6488	Japan	5.0	35	Small	2075	89	78
12	9995	Germany	3.0	26	Small	2330	109	100
13	11545	USA	1.0	20	Sporty	3320	305	170
14	9745	USA	1.0	27	Sporty	2885	153	100
15	12164	USA	1.0	19	Sporty	3310	302	225
16	11470	USA	3.0	30	Sporty	2695	133	110
17	9410	Japan	5.0	33	Sporty	2170	97	108
18	13945	Japan	5.0	27	Sporty	2710	125	140
19	13249	Japan	3.0	24	Sporty	2775	146	140
23	10565	USA	2.0	23	Compact	2640	151	110
24	10320	USA	1.0	26	Compact	2655	133	95
25	10945	USA	4.0	25	Compact	3065	181	141
26	9483	USA	2.0	24	Compact	2750	141	98
27	12145	Japan/USA	5.0	26	Compact	2920	132	125
28	12459	Japan/USA	4.0	24	Compact	2780	133	110
29	10989	Japan	5.0	25	Compact	2745	122	102
30	17879	Japan	4.0	21	Compact	3110	181	142
31	11650	Japan	5.0	21	Compact	2920	146	138
32	9995	USA	2.0	23	Compact	2645	151	110
34	11499	Japan/USA	5.0	23	Compact	2935	135	130
35	11588	Japan/USA	5.0	27	Compact	2920	122	115
36	18450	Sweden	3.0	23	Compact	2985	141	114

	Price	Country	Reliability	Mileage	Type	Weight	Disp.	HP
37	24760	Japan	5.0	20	Medium	3265	163	160
38	13150	USA	3.0	21	Medium	2880	151	110
39	12495	USA	2.0	22	Medium	2975	153	150
40	16342	USA	3.0	22	Medium	3450	202	147
41	15350	USA	2.0	22	Medium	3145	180	150
42	13195	USA	3.0	22	Medium	3190	182	140
43	14980	USA	1.0	23	Medium	3610	232	140
45	23300	Japan	5.0	21	Medium	3480	180	158
46	17899	Japan	5.0	22	Medium	3200	180	160
47	13150	USA	2.0	21	Medium	2765	151	110
49	21498	Japan	3.0	23	Medium	3480	180	190
50	16145	USA	3.0	23	Large	3325	231	165
51	14525	USA	1.0	18	Large	3855	305	170
52	17257	USA	3.0	20	Large	3850	302	150
54	15395	USA	3.0	18	Van	3735	202	150
55	12267	USA	3.0	18	Van	3665	182	145
56	14944	Japan	5.0	19	Van	3735	181	150

In [107]:
```
tre1 = tree.DecisionTreeClassifier('gini').fit(df1.ix[:,1:8],df1.Type)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-107-5a7803fcbe2f> in <module>()
----> 1 tre1 =
tree.DecisionTreeClassifier('gini').fit(df1.ix[:,1:8],df1.Type)

C:\Users\Administrator\Anaconda3\lib\site-packages\sklearn\tree\tree.py in fi
t(self, X, y, sample_weight, check_input, X_idx_sorted)
    737             sample_weight=sample_weight,
    738             check_input=check_input,
--> 739             X_idx_sorted=X_idx_sorted)
    740         return self
    741

C:\Users\Administrator\Anaconda3\lib\site-packages\sklearn\tree\tree.py in fi
t(self, X, y, sample_weight, check_input, X_idx_sorted)
    120         random_state = check_random_state(self.random_state)
    121         if check_input:
--> 122             X = check_array(X, dtype=DTYPE, accept_sparse="csc")
    123             y = check_array(y, ensure_2d=False, dtype=None)
    124             if issparse(X):

C:\Users\Administrator\Anaconda3\lib\site-packages\sklearn\utils\validation.p
y in check_array(array, accept_sparse, dtype, order, copy, force_all_finite,
 ensure_2d, allow_nd, ensure_min_samples, ensure_min_features, warn_on_dtype,
 estimator)
    380                                         force_all_finite)
    381     else:
--> 382         array = np.array(array, dtype=dtype, order=order, copy=copy)
    383
    384         if ensure_2d:

ValueError: could not convert string to float: 'Van'
```

In [99]:
```
dot_data = StringIO()
tree.export_graphviz(tre1, out_file=dot_data,
                     feature_names=col_names[1:7],
                     class_names=classnames,
                     filled=True,
                     rounded=True,
                     special_characters=True)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-99-e7bc95015e40> in <module>()
      1 dot_data = StringIO()
----> 2 tree.export_graphviz(tre1, out_file=dot_data,
      3                      feature_names=col_names[1:7],
      4                      class_names=classnames,
      5                      filled=True,

NameError: name 'tre1' is not defined
```