Hierarchical (Agglomerative) Clustering - Vehicle Data Set using Scipy

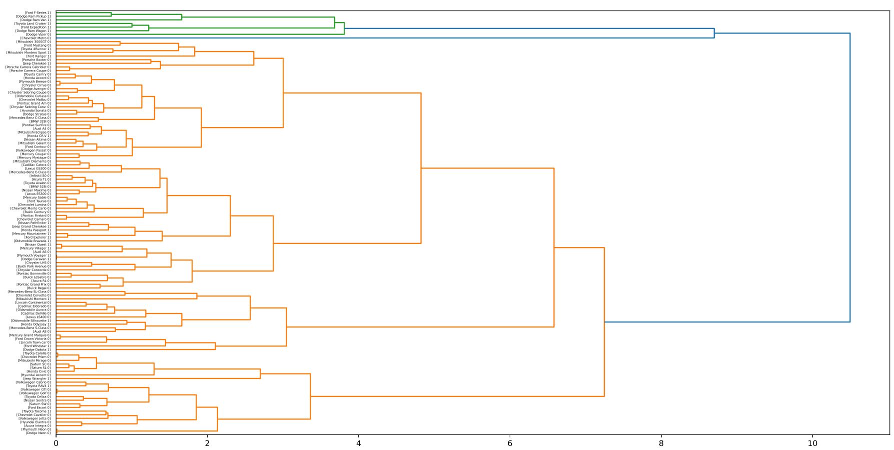
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from scipy.cluster import hierarchy
from scipy.spatial import distance_matrix
from sklearn import manifold,datasets
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import fcluster

## Load data ##
df=pd.read_csv('D:\Python\edx\Machine Learning\Clustering\cars_clus.csv')
with open('hierarchical_vehicle.txt','a') as f:
    print(df.head(),file=f)
    print(df.shape,file=f)

## Data Cleaning ## clear the dataset by dropping the rows that have null value:

with open('hierarchical_vehicle.txt','a') as f:
    print('Shape of data set before cleaning: ',df.size,file=f)

df[[ 'sales', 'resale', 'type', 'price', 'engine_s',
       'horsepow', 'wheelbas', 'width', 'length', 'curb_wgt', 'fuel_cap',
       'mpg', 'lnsales']] = df[['sales', 'resale', 'type', 'price', 'engine_s',
       'horsepow', 'wheelbas', 'width', 'length', 'curb_wgt', 'fuel_cap',
       'mpg', 'lnsales']].apply(pd.to_numeric, errors='coerce')

df=df.dropna()
df=df.reset_index(drop=True)
with open ('hierarchical_vehicle.txt','a') as f:
    print('Shape of the dataset after cleaning: ',df.size,file=f)
    print(df.head(5),file=f)

#Feature set
feat_set=df[['engine_s',  'horsepow', 'wheelbas', 'width', 'length', 'curb_wgt', 'fuel_ca

# Normalization - between 0,1 for each feature using MinMaxScalar

from sklearn.preprocessing import MinMaxScaler
x=feat_set.values
min_max=MinMaxScaler()
feat_matrix=min_max.fit_transform(x)
```

```
with open ('hierarchical_vehicle.txt','a') as f:
    print(feat_matrix[0:5],file=f)

# First method - Clustering using Scipy
##In agglomerative clustering, at each iteration, the algorithm must update the
# distance matrix to reflect the distance of the newly formed cluster with the remaining
# clusters in the forest. The following methods are supported in Scipy for calculating
# the distance between the newly formed cluster and each:
# - single - complete - average - weighted - centroid

import scipy
leng=feat_matrix.shape[0]
D=scipy.zeros([leng,leng])
for i in range(leng):
    for j in range(leng):
        D[i,j]=scipy.spatial.distance.euclidean(feat_matrix[i],feat_matrix[j])

import pylab
Z=hierarchy.linkage(D,'complete')

# for paritioning in clustering we draw a cutting line
max_d=3
clusters=fcluster(Z,max_d,criterion='distance')
k=5
clusters_max=fcluster(Z,k,criterion='maxclust')
with open ('hierarchical_vehicle.txt','a') as f:
    print(clusters,file=f)
    print(clusters_max,file=f)

# Dendrogram
fig = pylab.figure(figsize=(18,50))
def llf(id):
    return '[%s %s %s]' % (df['manufact'][id], df['model'][id], int(float(df['type'][id]

dendro=hierarchy.dendrogram(Z,leaf_label_func=llf, leaf_rotation=0, leaf_font_size =4, o

#Display plot
plt.show()
```

Solution:

```
  manufact    model   sales  resale   type   price engine_s horsepow wheelbas   width
length curb_wgt fuel_cap     mpg lnsales  partition
0     Acura  Integra  16.919  16.360  0.000  21.500    1.800  140.000  101.200  67.300
172.400    2.639   13.200  28.000   2.828       0.0
```

```
1    Acura        TL  39.384  19.875  0.000  28.400    3.200  225.000  108.100  70.300
192.900    3.517   17.200  25.000   3.673         0.0
2    Acura        CL  14.114  18.225  0.000  $null$    3.200  225.000  106.900  70.600
192.000    3.470   17.200  26.000   2.647         0.0
3    Acura        RL   8.588  29.725  0.000  42.000    3.500  210.000  114.600  71.400
196.600    3.850   18.000  22.000   2.150         0.0
4     Audi        A4  20.397  22.255  0.000  23.990    1.800  150.000  102.600  68.200
178.000    2.998   16.400  27.000   3.015         0.0
(159, 16)
Shape of data set before cleaning:  2544
Shape of the dataset after cleaning:  1872
  manufact    model   sales  resale  type  price  engine_s  horsepow  wheelbas  width
length  curb_wgt  fuel_cap   mpg  lnsales  partition
0    Acura  Integra  16.919  16.360   0.0  21.50       1.8     140.0     101.2
67.3   172.4     2.639      13.2  28.0    2.828        0.0
1    Acura       TL  39.384  19.875   0.0  28.40       3.2     225.0     108.1
70.3   192.9     3.517      17.2  25.0    3.673        0.0
2    Acura       RL   8.588  29.725   0.0  42.00       3.5     210.0     114.6
71.4   196.6     3.850      18.0  22.0    2.150        0.0
3     Audi       A4  20.397  22.255   0.0  23.99       1.8     150.0     102.6
68.2   178.0     2.998      16.4  27.0    3.015        0.0
4     Audi       A6  18.780  23.555   0.0  33.95       2.8     200.0     108.7
76.1   192.0     3.561      18.5  22.0    2.933        0.0
[[0.11428571 0.21518987 0.18655098 0.28143713 0.30625832 0.2310559
  0.13364055 0.43333333]
 [0.31428571 0.43037975 0.3362256  0.46107784 0.5792277  0.50372671
  0.31797235 0.33333333]
 [0.35714286 0.39240506 0.47722343 0.52694611 0.62849534 0.60714286
  0.35483871 0.23333333]
 [0.11428571 0.24050633 0.21691974 0.33532934 0.38082557 0.34254658
  0.28110599 0.4       ]
 [0.25714286 0.36708861 0.34924078 0.80838323 0.56724368 0.5173913
  0.37788018 0.23333333]]
[ 1  5  5  6  5  4  6  5  5  5  5  5  4  4  5  1  6  5  5  5  4  2 11  6
  6  5  6  5  1  6  6 10  9  8  9  3  5  1  7  6  5  3  5  3  8  7  9  2
  6  6  5  4  2  1  6  5  2  7  5  5  5  4  4  3  2  6  6  5  7  4  7  6
  6  5  3  5  5  6  5  4  4  1  6  5  5  5  6  4  5  4  1  6  5  6  6  5
  5  5  7  7  7  2  2  1  2  6  5  1  1  1  7  8  1  1  6  1  1]
[1 3 3 3 3 2 3 3 3 3 3 3 2 2 3 1 3 3 3 3 2 1 5 3 3 3 3 3 1 3 3 4 4 4 4 2 3
 1 3 3 3 2 3 2 4 3 4 1 3 3 3 2 1 1 3 3 1 3 3 3 3 2 2 2 1 3 3 3 3 2 3 3 3 3
 2 3 3 3 3 2 2 1 3 3 3 3 3 2 3 2 1 3 3 3 3 3 3 3 3 3 3 1 1 1 1 3 3 1 1 1 3
 4 1 1 3 1 1]
```



Hierarchical (Agglomerative) Clustering - Vehicle Data Set using Scikit-learn

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from scipy.cluster import hierarchy
from scipy.spatial import distance_matrix
from sklearn import manifold,datasets
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import fcluster

## Load data ##
df=pd.read_csv('D:\Python\edx\Machine Learning\Clustering\cars_clus.csv')
with open('hierarchical_vehicle_s.txt','a') as f:
    print(df.head(),file=f)
    print(df.shape,file=f)

## Data Cleaning ## clear the dataset by dropping the rows that have null value:

with open('hierarchical_vehicle_s.txt','a') as f:
    print('Shape of data set before cleaning: ',df.size,file=f)

df[[ 'sales', 'resale', 'type', 'price', 'engine_s',
```

```
        'horsepow', 'wheelbas', 'width', 'length', 'curb_wgt', 'fuel_cap',
        'mpg', 'lnsales']] = df[['sales', 'resale', 'type', 'price', 'engine_s',
        'horsepow', 'wheelbas', 'width', 'length', 'curb_wgt', 'fuel_cap',
        'mpg', 'lnsales']].apply(pd.to_numeric, errors='coerce')

df=df.dropna()
df=df.reset_index(drop=True)
with open ('hierarchical_vehicle_s.txt','a') as f:
    print('Shape of the dataset after cleaning: ',df.size,file=f)
    print(df.head(5),file=f)

#Feature set
feat_set=df[['engine_s',  'horsepow', 'wheelbas', 'width', 'length', 'curb_wgt', 'fuel_c

# Normalization - between 0,1 for each feature using MinMaxScalar

from sklearn.preprocessing import MinMaxScaler
x=feat_set.values
min_max=MinMaxScaler()
feat_matrix=min_max.fit_transform(x)
with open ('hierarchical_vehicle_s.txt','a') as f:
    print(feat_matrix[0:5],file=f)

# Second method - Clustering using scikit-learn
d_mat=distance_matrix(feat_matrix,feat_matrix)
with open ('hierarchical_vehicle_s.txt','a') as f:
    print(d_mat,file=f)

#AgglomerativeClustering performs a hierarchical clustering using a bottom up approach.
# The linkage criteria determines the metric used for the merge strategy:
# Ward minimizes the sum of squared differences within all clusters. It is a variance-mi
# approach and in this sense is similar to the k-means objective function but tackled wi
# agglomerative hierarchical approach.
# Maximum or complete linkage minimizes the maximum distance between observations of pai
# Average linkage minimizes the average of the distances between all observations of pai

agglom=AgglomerativeClustering(n_clusters=6,linkage='complete')
agglom.fit(feat_matrix)
with open ('hierarchical_vehicle_s.txt','a') as f:
    print(agglom.labels_,file=f)

# Adding new column - cluster to the data
df['cluster_']=agglom.labels_
with open ('hierarchical_vehicle_s.txt','a') as f:
    print(df.head(),file=f)
```

```
## Plotting scatter plot for data points with their clusters
import matplotlib.cm as cm
n_clusters=max(agglom.labels_)+1
colors=cm.rainbow(np.linspace(0,1,n_clusters))
cluster_labels=list(range(0,n_clusters))

plt.figure(figsize=(16,14))
for color, label in zip(colors,cluster_labels):
    subset=df[df.cluster_==label]
    for i in subset.index:
        plt.text(subset.horsepow[i], subset.mpg[i],str(subset['model'][i]), rotation=25)
    plt.scatter(subset.horsepow, subset.mpg, s= subset.price*10, c=color, label='cluster
#    plt.scatter(subset.horsepow, subset.mpg)
plt.legend()
plt.title('Clusters')
plt.xlabel('horsepow')
plt.ylabel('mpg')

 # Centroids of each cluster is not clear in scatter plot, so we can summarize first
 #classes and then the clusters. There are two classes - Cars and Trucks

qdf=df.groupby(['cluster_','type'])['cluster_'].count()
with open ('hierarchical_vehicle_s.txt','a') as f:
    print(qdf,file=f)

#For characteristics of each cluster
agg_cars=df.groupby(['cluster_','type'])['horsepow','engine_s','mpg','price'].mean()
with open ('hierarchical_vehicle_s.txt','a') as f:
    print(agg_cars,file=f)

##It is obvious that we have 3 main clusters with the majority of vehicles in those.
##Cars:
    ##Cluster 1: with almost high mpg, and low in horsepower.
    ##Cluster 2: with good mpg and horsepower, but higher price than average.
   ## Cluster 3: with low mpg, high horsepower, highest price.
##Trucks:
    ##Cluster 1: with almost highest mpg among trucks, and lowest in horsepower and price
    ##Cluster 2: with almost low mpg and medium horsepower, but higher price than averag
    ##Cluster 3: with good mpg and horsepower, low price.

plt.figure(figsize=(16,10))
for color, label in zip(colors, cluster_labels):
    subset = agg_cars.loc[(label,),]
    for i in subset.index:
        plt.text(subset.loc[i][0]+5, subset.loc[i][2], 'type='+str(int(i)) + ', price='+
    plt.scatter(subset.horsepow, subset.mpg, s=subset.price*20, c=color, label='cluster'+
```

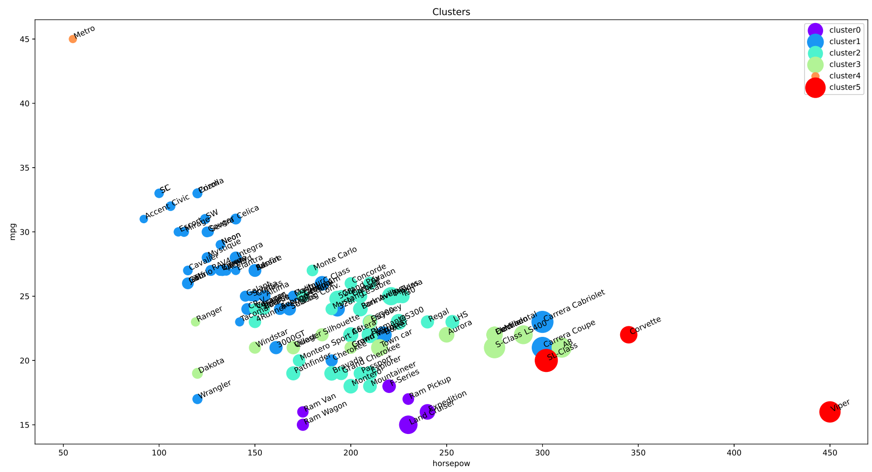
```
plt.legend()
plt.title('Clusters')
plt.xlabel('horsepow')
plt.ylabel('mpg')

#Display plot
plt.show()
```

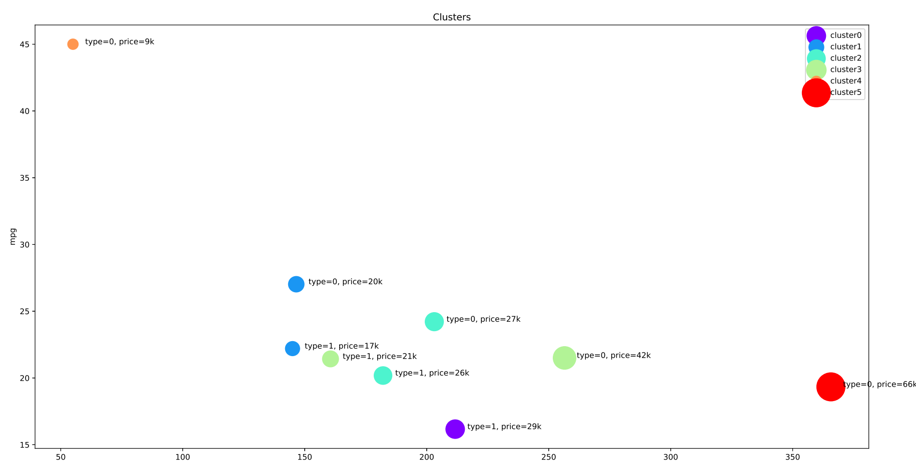
Solution:

```
  manufact    model   sales  resale   type   price engine_s horsepow wheelbas   width
length curb_wgt fuel_cap     mpg lnsales  partition
0    Acura  Integra  16.919  16.360  0.000  21.500    1.800  140.000  101.200  67.300
172.400    2.639   13.200  28.000   2.828        0.0
1    Acura       TL  39.384  19.875  0.000  28.400    3.200  225.000  108.100  70.300
192.900    3.517   17.200  25.000   3.673        0.0
2    Acura       CL  14.114  18.225  0.000  $null$    3.200  225.000  106.900  70.600
192.000    3.470   17.200  26.000   2.647        0.0
3    Acura       RL   8.588  29.725  0.000  42.000    3.500  210.000  114.600  71.400
196.600    3.850   18.000  22.000   2.150        0.0
4     Audi       A4  20.397  22.255  0.000  23.990    1.800  150.000  102.600  68.200
178.000    2.998   16.400  27.000   3.015        0.0
(159, 16)
Shape of data set before cleaning:  2544
Shape of the dataset after cleaning:  1872
  manufact    model   sales  resale  type  price  engine_s  horsepow  wheelbas  width
length  curb_wgt  fuel_cap   mpg  lnsales  partition
0    Acura  Integra  16.919  16.360   0.0  21.50       1.8     140.0     101.2
67.3   172.4     2.639      13.2  28.0    2.828        0.0
1    Acura       TL  39.384  19.875   0.0  28.40       3.2     225.0     108.1
70.3   192.9     3.517      17.2  25.0    3.673        0.0
2    Acura       RL   8.588  29.725   0.0  42.00       3.5     210.0     114.6
71.4   196.6     3.850      18.0  22.0    2.150        0.0
3     Audi       A4  20.397  22.255   0.0  23.99       1.8     150.0     102.6
68.2   178.0     2.998      16.4  27.0    3.015        0.0
4     Audi       A6  18.780  23.555   0.0  33.95       2.8     200.0     108.7
76.1   192.0     3.561      18.5  22.0    2.933        0.0
[[0.11428571 0.21518987 0.18655098 0.28143713 0.30625832 0.2310559
  0.13364055 0.43333333]
 [0.31428571 0.43037975 0.3362256  0.46107784 0.5792277  0.50372671
  0.31797235 0.33333333]
 [0.35714286 0.39240506 0.47722343 0.52694611 0.62849534 0.60714286
  0.35483871 0.23333333]
 [0.11428571 0.24050633 0.21691974 0.33532934 0.38082557 0.34254658
  0.28110599 0.4       ]
```

```
 [0.25714286 0.36708861 0.34924078 0.80838323 0.56724368 0.5173913
  0.37788018 0.23333333]]
[[0.         0.57777143 0.75455727 ... 0.28530295 0.24917241 0.18879995]
 [0.57777143 0.         0.22798938 ... 0.36087756 0.66346677 0.62201282]
 [0.75455727 0.22798938 0.         ... 0.51727787 0.81786095 0.77930119]
 ...
 [0.28530295 0.36087756 0.51727787 ... 0.         0.41797928 0.35720492]
 [0.24917241 0.66346677 0.81786095 ... 0.41797928 0.         0.15212198]
 [0.18879995 0.62201282 0.77930119 ... 0.35720492 0.15212198 0.        ]]
[1 2 2 1 2 3 1 2 2 2 2 2 3 3 2 1 1 2 2 2 5 1 4 1 1 2 1 2 1 1 1 5 0 0 0 3 2
 1 2 1 2 3 2 3 0 3 0 1 1 1 2 3 1 1 1 2 1 1 2 2 2 3 3 3 1 1 1 2 1 2 2 1 1 2
 3 2 3 1 2 3 5 1 1 2 3 2 1 3 2 3 1 1 2 1 1 2 2 2 1 1 1 1 1 1 1 1 2 1 1 1 2
 0 1 1 1 1 1]
  manufact    model   sales  resale  type  price  engine_s  horsepow  wheelbas  width
length  curb_wgt  fuel_cap   mpg  lnsales  partition  cluster_
0    Acura  Integra  16.919  16.360   0.0  21.50       1.8     140.0     101.2
67.3   172.4     2.639      13.2  28.0    2.828        0.0         1
1    Acura       TL  39.384  19.875   0.0  28.40       3.2     225.0     108.1
70.3   192.9     3.517      17.2  25.0    3.673        0.0         2
2    Acura       RL   8.588  29.725   0.0  42.00       3.5     210.0     114.6
71.4   196.6     3.850      18.0  22.0    2.150        0.0         2
3     Audi       A4  20.397  22.255   0.0  23.99       1.8     150.0     102.6
68.2   178.0     2.998      16.4  27.0    3.015        0.0         1
4     Audi       A6  18.780  23.555   0.0  33.95       2.8     200.0     108.7
76.1   192.0     3.561      18.5  22.0    2.933        0.0         2
cluster_  type
0         1.0      6
1         0.0     47
          1.0      5
2         0.0     27
          1.0     11
3         0.0     10
          1.0      7
4         0.0      1
5         0.0      3
Name: cluster_, dtype: int64
                  horsepow  engine_s        mpg      price
cluster_ type
0        1.0   211.666667  4.483333  16.166667  29.024667
1        0.0   146.531915  2.246809  27.021277  20.306128
         1.0   145.000000  2.580000  22.200000  17.009200
2        0.0   203.111111  3.303704  24.214815  27.750593
         1.0   182.090909  3.345455  20.181818  26.265364
3        0.0   256.500000  4.410000  21.500000  42.870400
         1.0   160.571429  3.071429  21.428571  21.527714
4        0.0    55.000000  1.000000  45.000000   9.235000
```

5	0.0	365.666667	6.233333	19.333333	66.010000


Clusters
cluster0
cluster1
cluster2
cluster3
cluster4
cluster5
mpg
horsepow


Clusters
cluster0
cluster1
cluster2
cluster3
cluster4
cluster5
type=0, price=9k
type=0, price=20k
type=0, price=27k
type=1, price=17k
type=1, price=21k
type=0, price=42k
type=1, price=26k
type=0, price=66k
type=1, price=29k
mpg
horsepow