In [1]:
```
import seaborn as sns
import pandas as pd

from sklearn.preprocessing import MinMaxScaler

from sklearn.cluster import KMeans

from sklearn.metrics import confusion_matrix

%matplotlib inline
```

In [2]:
```
scores=pd.read_csv("2013-2017_School_Math_Results_-_All.csv",na_values="s")
scores.head()
```

Out[2]:

	DBN	School Name	Grade	Year	Category	Number Tested	Mean Scale Score	Level1_N	Level1_%	Lev
0	01M015	PS 015 ROBERTO CLEMENTE	3	2013	All Students	27	278.0	16.0	59.3	11.(
1	01M015	PS 015 ROBERTO CLEMENTE	3	2014	All Students	18	286.0	6.0	33.3	9.0
2	01M015	PS 015 ROBERTO CLEMENTE	3	2015	All Students	17	280.0	10.0	58.8	4.0
3	01M015	PS 015 ROBERTO CLEMENTE	3	2016	All Students	21	275.0	13.0	61.9	4.0
4	01M015	PS 015 ROBERTO CLEMENTE	3	2017	All Students	29	302.0	8.0	27.6	9.0

In [3]:
```
scores.dropna(inplace=True)
```

In [4]:
```
all_grades_filter=scores["Grade"]=="All Grades"
```

In [5]:
```
scores2=scores[~all_grades_filter]
```

In [6]:
```
scores2.head()
```

Out[6]:

	DBN	School Name	Grade	Year	Category	Number Tested	Mean Scale Score	Level1_N	Level1_%	Lev
0	01M015	PS 015 ROBERTO CLEMENTE	3	2013	All Students	27	278.0	16.0	59.3	11.(
1	01M015	PS 015 ROBERTO CLEMENTE	3	2014	All Students	18	286.0	6.0	33.3	9.0
2	01M015	PS 015 ROBERTO CLEMENTE	3	2015	All Students	17	280.0	10.0	58.8	4.0
3	01M015	PS 015 ROBERTO CLEMENTE	3	2016	All Students	21	275.0	13.0	61.9	4.0
4	01M015	PS 015 ROBERTO CLEMENTE	3	2017	All Students	29	302.0	8.0	27.6	9.0

In [7]:
```
scores3=scores2[["Number Tested","Mean Scale Score","Level1_N","Level1_%","Lev
el2_N","Level2_%","Level3_N","Level3_%","Level4_N","Level4_%","Level3+4_N","Le
vel3+4_%"]]
scores3.head()
```

Out[7]:

	Number Tested	Mean Scale Score	Level1_N	Level1_%	Level2_N	Level2_%	Level3_N	Level3_%	Level4
0	27	278.0	16.0	59.3	11.0	40.7	0.0	0.0	0.0
1	18	286.0	6.0	33.3	9.0	50.0	2.0	11.1	1.0
2	17	280.0	10.0	58.8	4.0	23.5	2.0	11.8	1.0
3	21	275.0	13.0	61.9	4.0	19.0	4.0	19.0	0.0
4	29	302.0	8.0	27.6	9.0	31.0	7.0	24.1	5.0

In [8]:
```
x=scores3[["Level1_N","Level1_%","Level2_N","Level2_%"]]
x.head()
```

Out[8]:

	Level1_N	Level1_%	Level2_N	Level2_%
0	16.0	59.3	11.0	40.7
1	6.0	33.3	9.0	50.0
2	10.0	58.8	4.0	23.5
3	13.0	61.9	4.0	19.0
4	8.0	27.6	9.0	31.0

In [9]:
```
kmeans = KMeans(n_clusters=4)
```

In [10]:
```
kmeans.fit(x)
```

Out[10]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=4, n_init=10, n_jobs=None, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

In [11]:
```
clusters = kmeans.predict(x)
```

In [12]:
```
clusters
```

Out[12]:
```
array([3, 1, 3, ..., 0, 3, 3], dtype=int32)
```

In [13]:
```
scores3["clusters"]=clusters
scores3.head()
```

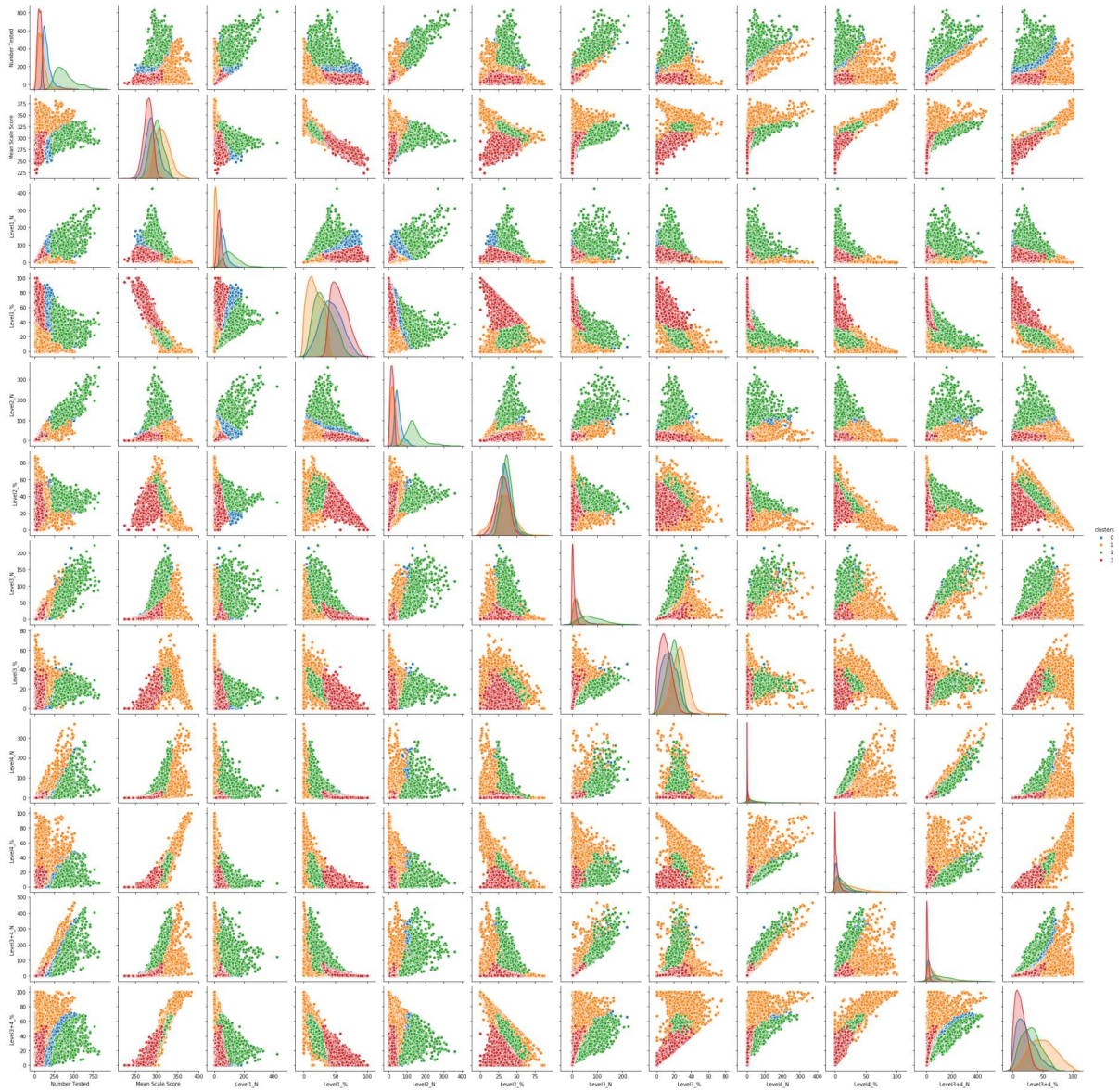
```
/usr/local/lib/python3.4/site-packages/ipykernel_launcher.py:1: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
```

Out[13]:

	Number Tested	Mean Scale Score	Level1_N	Level1_%	Level2_N	Level2_%	Level3_N	Level3_%	Level4
0	27	278.0	16.0	59.3	11.0	40.7	0.0	0.0	0.0
1	18	286.0	6.0	33.3	9.0	50.0	2.0	11.1	1.0
2	17	280.0	10.0	58.8	4.0	23.5	2.0	11.8	1.0
3	21	275.0	13.0	61.9	4.0	19.0	4.0	19.0	0.0
4	29	302.0	8.0	27.6	9.0	31.0	7.0	24.1	5.0

In [14]:
```
sns.pairplot(scores3,hue="clusters")
```

Out[14]: <seaborn.axisgrid.PairGrid at 0x7fac9c6c6f28>



In [15]:
```
x2=scores3[["Level3_N","Level3_%","Level4_N","Level4_%"]]
x2.head()
```

Out[15]:

	Level3_N	Level3_%	Level4_N	Level4_%
0	0.0	0.0	0.0	0.0
1	2.0	11.1	1.0	5.6
2	2.0	11.8	1.0	5.9
3	4.0	19.0	0.0	0.0
4	7.0	24.1	5.0	17.2

In [16]:
```
kmeans2 = KMeans(n_clusters=5)
```

In [17]:
```
kmeans2.fit(x)
```

Out[17]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=5, n_init=10, n_jobs=None, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

In [18]:
```
clusters2 = kmeans2.predict(x)
```

In [19]:
```
clusters2
```

Out[19]:
```
array([0, 2, 0, ..., 1, 0, 0], dtype=int32)
```

In [20]:
```
scores3["clusters2"]=clusters2
scores3.head()
```

```
/usr/local/lib/python3.4/site-packages/ipykernel_launcher.py:1: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
```

Out[20]:

	Number Tested	Mean Scale Score	Level1_N	Level1_%	Level2_N	Level2_%	Level3_N	Level3_%	Level4
0	27	278.0	16.0	59.3	11.0	40.7	0.0	0.0	0.0
1	18	286.0	6.0	33.3	9.0	50.0	2.0	11.1	1.0
2	17	280.0	10.0	58.8	4.0	23.5	2.0	11.8	1.0
3	21	275.0	13.0	61.9	4.0	19.0	4.0	19.0	0.0
4	29	302.0	8.0	27.6	9.0	31.0	7.0	24.1	5.0

In [21]:
```
sns.pairplot(scores3,hue="clusters2")
```

```
/usr/local/lib/python3.4/site-packages/seaborn/distributions.py:290: UserWarn
ing: Data must have variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
```

Out[21]: <seaborn.axisgrid.PairGrid at 0x7fac998349e8>