—

K Means Clustering Project

For this project we will attempt to use KMeans Clustering to cluster Universities into to two groups, Private and Public.

Note: We actually have the labels for this data set, but we will NOT use them for the KMeans clustering algorithm, since that is an unsupervised learning algorithm.

When using the KMeans algorithm under normal circumstances, we won't have labels. In this case we will use the labels to try to get an idea of how well the algorithm performed using the classification report and confusion matrix.

The Data

We will use a data frame with 777 observations on the following 18 variables.

- Private A factor with levels No and Yes indicating private or public university
- Apps Number of applications received
- Accept Number of applications accepted
- Enroll Number of new students enrolled
- Top10perc Pct. new students from top 10% of H.S. class
- Top25perc Pct. new students from top 25% of H.S. class
- F.Undergrad Number of fulltime undergraduates
- P.Undergrad Number of parttime undergraduates
- Outstate Out-of-state tuition
- Room.Board Room and board costs
- Books Estimated book costs
- Personal Estimated personal spending
- PhD Pct. of faculty with Ph.D.’s
- Terminal Pct. of faculty with terminal degree
- S.F.Ratio Student/faculty ratio
- perc.alumni Pct. alumni who donate
- Expend Instructional expenditure per student
- Grad.Rate Graduation rate

Import Libraries

-

Import the libraries you usually use for data analysis.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Get the Data

Read in the College_Data file using read_csv. Figure out how to set the first column as the index.

In [2]:
```
df = pd.read_csv('College_Data', index_col=0)
df.head()
```

Out[2]:

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outst
Abilene Christian University	Yes	1660	1232	721	23	52	2885	537	7
Adelphi University	Yes	2186	1924	512	16	29	2683	1227	12
Adrian College	Yes	1428	1097	336	22	50	1036	99	11
Agnes Scott College	Yes	417	349	137	60	89	510	63	12
Alaska Pacific University	Yes	193	146	55	16	44	249	869	7

In [3]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 777 entries, Abilene Christian University to York College of Pe
nnsylvania
Data columns (total 18 columns):
Private        777 non-null object
Apps           777 non-null int64
Accept         777 non-null int64
Enroll         777 non-null int64
Top10perc      777 non-null int64
Top25perc      777 non-null int64
F.Undergrad    777 non-null int64
P.Undergrad    777 non-null int64
Outstate       777 non-null int64
Room.Board     777 non-null int64
Books          777 non-null int64
Personal       777 non-null int64
PhD            777 non-null int64
Terminal       777 non-null int64
S.F.Ratio      777 non-null float64
perc.alumni    777 non-null int64
Expend         777 non-null int64
Grad.Rate      777 non-null int64
dtypes: float64(1), int64(16), object(1)
memory usage: 115.3+ KB
```

In [4]:
```
1 df.describe()
```

Out[4]:

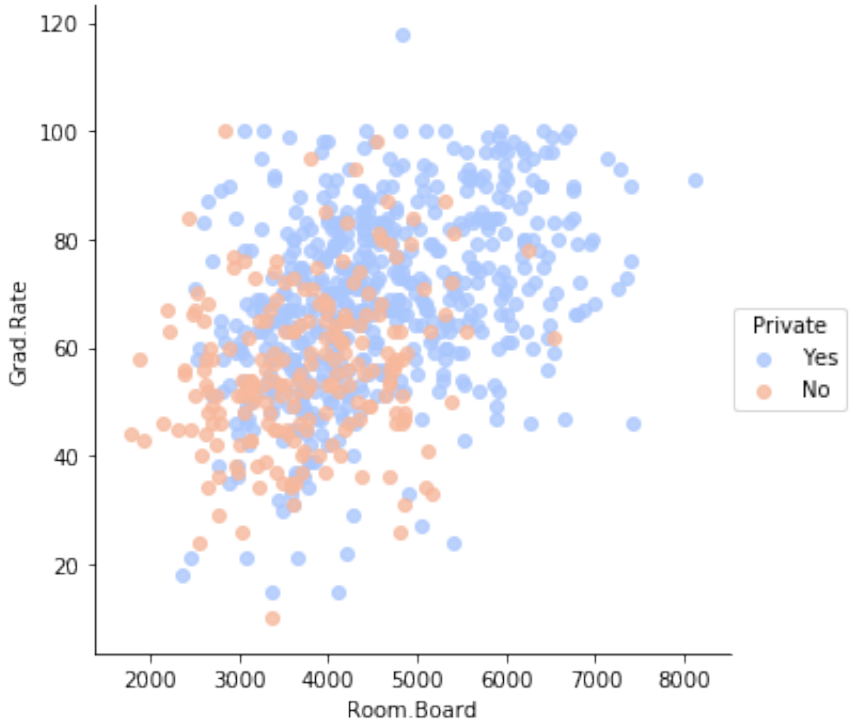
	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Under
count	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.00
mean	3001.638353	2018.804376	779.972973	27.558559	55.796654	3699.907336	855.29
std	3870.201484	2451.113971	929.176190	17.640364	19.804778	4850.420531	1522.43
min	81.000000	72.000000	35.000000	1.000000	9.000000	139.000000	1.00
25%	776.000000	604.000000	242.000000	15.000000	41.000000	992.000000	95.00
50%	1558.000000	1110.000000	434.000000	23.000000	54.000000	1707.000000	353.00
75%	3624.000000	2424.000000	902.000000	35.000000	69.000000	4005.000000	967.00
max	48094.000000	26330.000000	6392.000000	96.000000	100.000000	31643.000000	21836.00

Exploratory Data Analysis

Create a scatterplot of Grad.Rate versus Room.Board where the points are colored by the Private column.

In [6]:
```
sns.lmplot(data = df, x = 'Room.Board', y = 'Grad.Rate',
           hue = 'Private', palette = 'coolwarm', fit_reg = False)
```

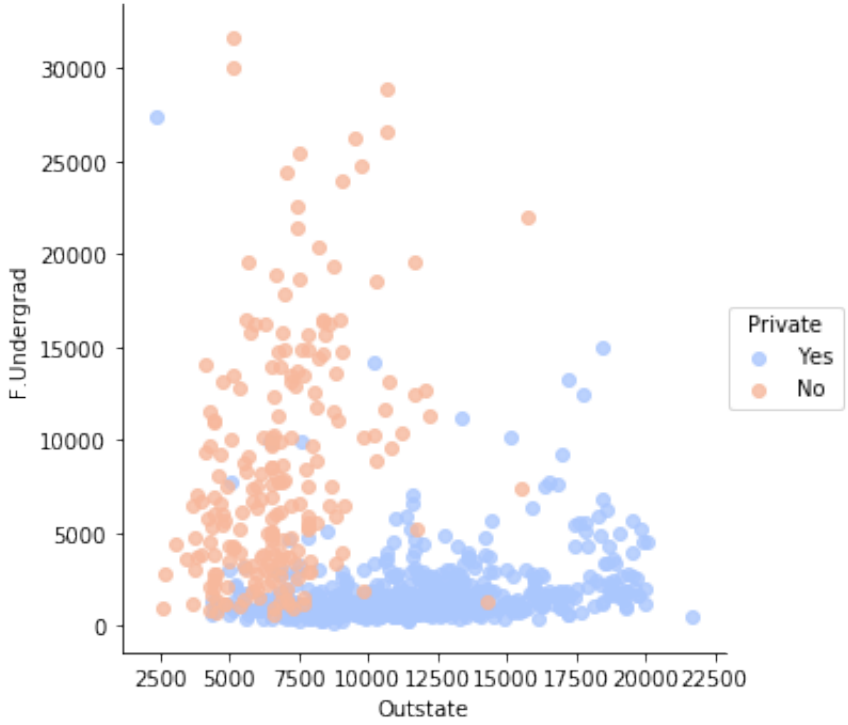
Out[6]: <seaborn.axisgrid.FacetGrid at 0x1a1af66a90>



Create a scatterplot of F.Undergrad versus Outstate where the points are colored by the Private column.

In [7]:
```
sns.lmplot(data = df, x = 'Outstate', y = 'F.Undergrad',
           hue = 'Private', palette = 'coolwarm', fit_reg = False)
```

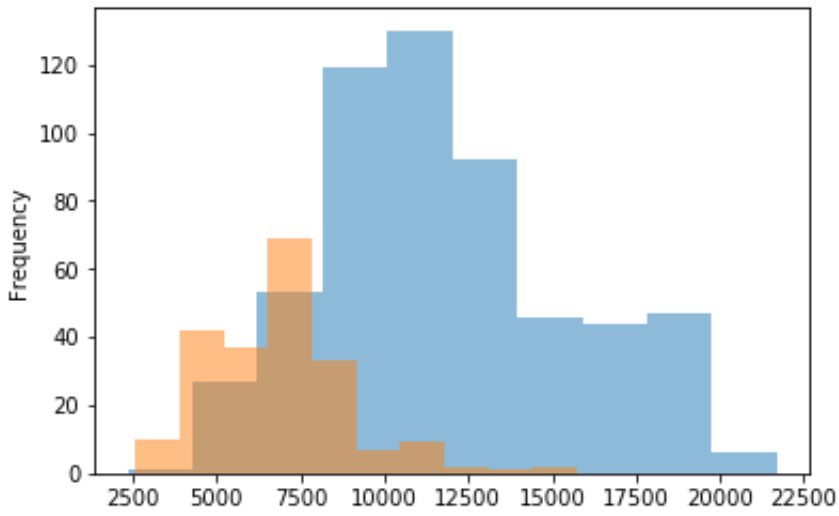
Out[7]: <seaborn.axisgrid.FacetGrid at 0x1a1af9b358>



Create a stacked histogram showing Out of State Tuition based on the Private column.

In [8]:
```
df[df['Private'] == 'Yes']['Outstate'].plot(kind = 'hist', alpha = 0.
df[df['Private'] == 'No']['Outstate'].plot(kind = 'hist', alpha = 0.5
```

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1ecd6278>

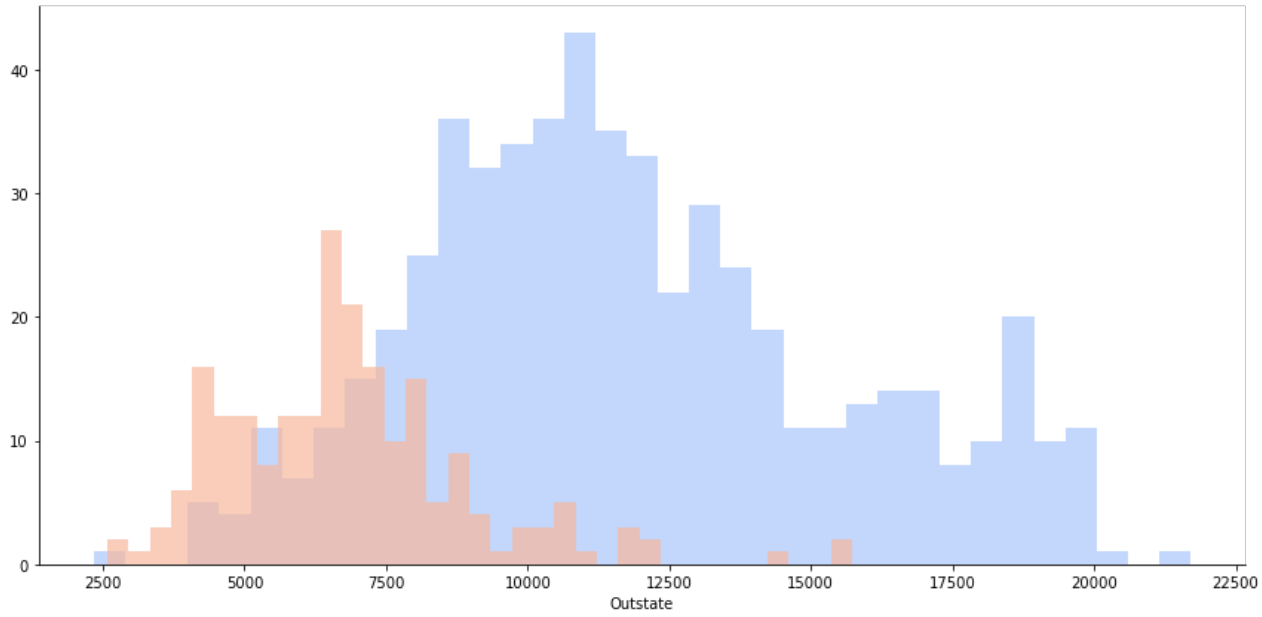


Same graph as above using [sns.FacetGrid]

Ref: https://stanford.edu/~mwaskom/software/seaborn/generated/seaborn.FacetGrid.html (https://stanford.edu/~mwaskom/software/seaborn/generated/seaborn.FacetGrid.html)

In [7]:
```
g = sns.FacetGrid(df, hue = 'Private', palette = 'coolwarm',
                  size = 6, aspect = 2)
g.map(plt.hist, 'Outstate', bins = 35, alpha = 0.7)
```

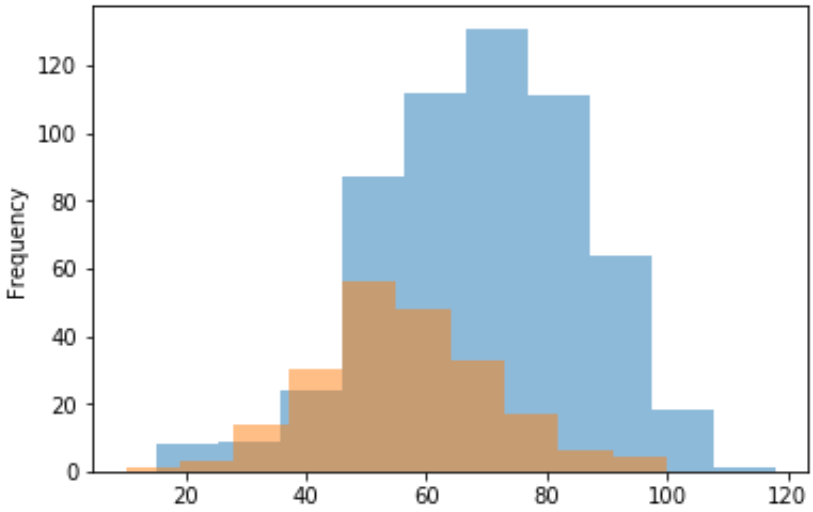
Out[7]: <seaborn.axisgrid.FacetGrid at 0x1a231d1860>



Create a similar histogram for the Grad.Rate column.

In [10]:
```
df[df['Private'] == 'Yes']['Grad.Rate'].plot(kind = 'hist', alpha = 0
df[df['Private'] == 'No']['Grad.Rate'].plot(kind = 'hist', alpha = 0.
```

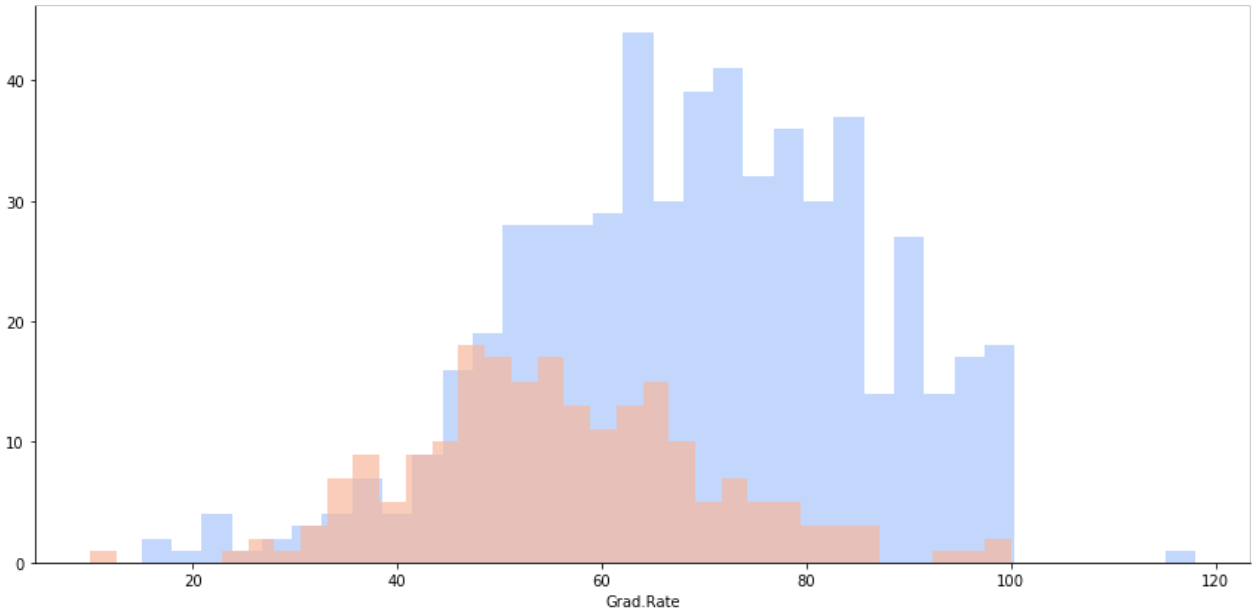
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1f2105f8>



Same graph as above using sns.FacetGrid

In [11]:
```
g = sns.FacetGrid(df, hue = 'Private', palette = 'coolwarm',
                  size = 6, aspect = 2)
g.map(plt.hist, 'Grad.Rate', bins = 35, alpha = 0.7)
```

Out[11]: <seaborn.axisgrid.FacetGrid at 0x1a1afbdc50>



Notice how there seems to be a private school with a graduation rate of higher than 100%.

Let's find out the name of the school.

In [12]:
```
1 df[df['Grad.Rate'] > 100]
```

Out[12]:

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outs
Cazenovia College	Yes	3847	3433	527	9	35	1010	12	9

Let's set that school's graduation rate to 100 so it makes sense.

In [13]:
```
1 df['Grad.Rate']['Cazenovia College'] = 100
```

```
/Users/Jayashri/anaconda/lib/python3.6/site-packages/ipykernel_launche
r.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

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

Let's verify if there is any school with graduation rate higher than 100%.

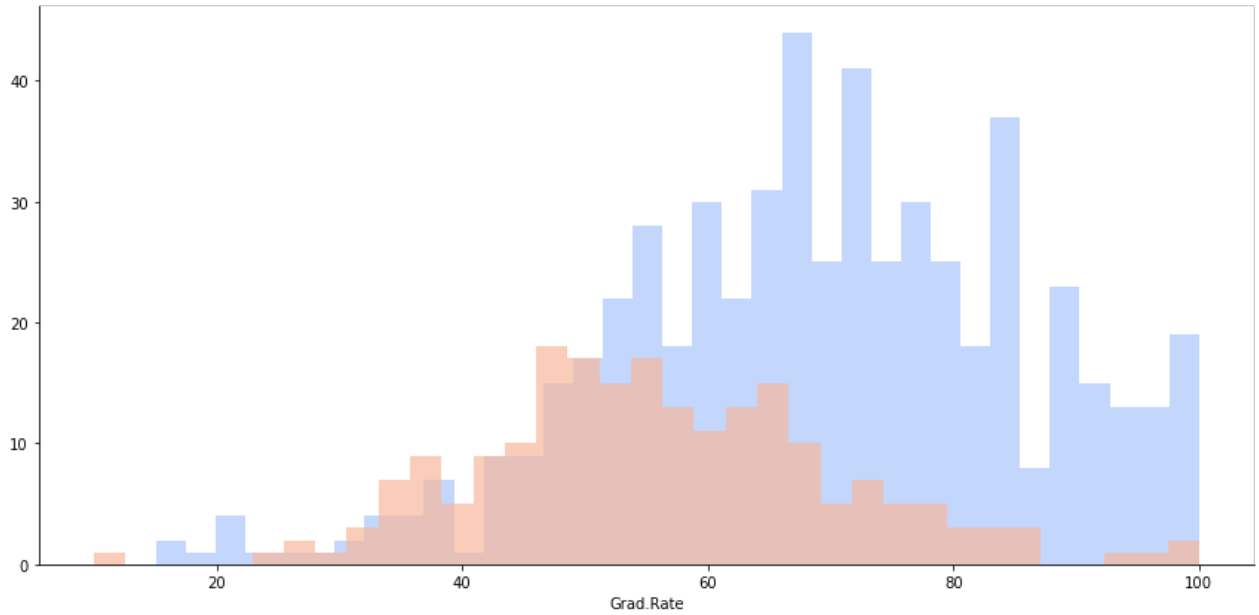
In [14]:
```
1 df[df['Grad.Rate'] > 100]
```

Out[14]:

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Roo

In [15]:
```
g = sns.FacetGrid(df, hue = 'Private', palette = 'coolwarm',
                  size = 6, aspect = 2)
g.map(plt.hist, 'Grad.Rate', bins = 35, alpha = 0.7)
```

Out[15]: <seaborn.axisgrid.FacetGrid at 0x1a1f3092e8>



K Means Cluster Creation

Now it is time to create the Cluster labels!

Import KMeans from SciKit Learn.

In [9]:
```
from sklearn.cluster import KMeans
```

Create an instance of a K Means model with 2 clusters.

In [10]:
```
kmeans = KMeans(n_clusters = 2)
```

Fit the model to all the data except for the Private label.

In [18]:
```
kmeans.fit(df.drop('Private', axis = 1))
```

Out[18]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=2, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

Let's look at the cluster center vectors

Let's look at the cluster center vectors

In [19]:
```
kmeans.cluster_centers_
```

Out[19]:
```
array([[  1.03631389e+04,   6.55089815e+03,   2.56972222e+03,
          4.14907407e+01,   7.02037037e+01,   1.30619352e+04,
          2.46486111e+03,   1.07191759e+04,   4.64347222e+03,
          5.95212963e+02,   1.71420370e+03,   8.63981481e+01,
          9.13333333e+01,   1.40277778e+01,   2.00740741e+01,
          1.41705000e+04,   6.75925926e+01],
       [  1.81323468e+03,   1.28716592e+03,   4.91044843e+02,
          2.53094170e+01,   5.34708520e+01,   2.18854858e+03,
          5.95458894e+02,   1.03957085e+04,   4.31136472e+03,
          5.41982063e+02,   1.28033632e+03,   7.04424514e+01,
          7.78251121e+01,   1.40997010e+01,   2.31748879e+01,
          8.93204634e+03,   6.50926756e+01]])
```

Evaluation

There is really no way to evaluate a cluster in real life. However, we do have the labels, so we can evaluate our clusters.

Create a new column for df called 'Cluster', which is a 1 for a Private school, and a 0 for a public school.

In [11]:
```
def converter(x):
    if x == 'Yes':
        return 1
    else:
        return 0

df['Cluster'] = df['Private'].apply(converter)
```

In [12]:
```
1 df.head()
```

Out[12]:

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outst
Abilene Christian University	Yes	1660	1232	721	23	52	2885	537	7
Adelphi University	Yes	2186	1924	512	16	29	2683	1227	12
Adrian College	Yes	1428	1097	336	22	50	1036	99	11
Agnes Scott College	Yes	417	349	137	60	89	510	63	12
Alaska Pacific University	Yes	193	146	55	16	44	249	869	7

Create a confusion matrix and classification report to see how well the Kmeans clustering worked without being given any labels.

In [123]:
```
1
```

```
[[138  74]
 [531  34]]
             precision    recall  f1-score   support

          0       0.21      0.65      0.31       212
          1       0.31      0.06      0.10       565

avg / total       0.29      0.22      0.16       777
```

In [22]:
```
from sklearn.metrics import confusion_matrix, classification_report

print(confusion_matrix(df['Cluster'], kmeans.labels_))
print(classification_report(df['Cluster'], kmeans.labels_))
```

```
[[ 74 138]
 [ 34 531]]
             precision    recall  f1-score   support

          0       0.69      0.35      0.46       212
          1       0.79      0.94      0.86       565

avg / total       0.76      0.78      0.75       777
```

In [31]:
```
df['Predictions'] = kmeans.labels_
print(df[['Cluster', 'Predictions']].head())
```

```
                              Cluster  Predictions
Abilene Christian University        1            1
Adelphi University                  1            1
Adrian College                      1            1
Agnes Scott College                 1            1
Alaska Pacific University           1            1
```

Let's examine the data that got mislabeled.

In [33]:
```
1 df[df['Cluster'] != df['Predictions']].count()
```

Out[33]:
```
Private        172
Apps           172
Accept         172
Enroll         172
Top10perc      172
Top25perc      172
F.Undergrad    172
P.Undergrad    172
Outstate       172
Room.Board     172
Books          172
Personal       172
PhD            172
Terminal       172
S.F.Ratio      172
perc.alumni    172
Expend         172
Grad.Rate      172
Cluster        172
Predictions    172
dtype: int64
```

In [35]:
```
1 df[df['Cluster'] != df['Predictions']]
```

Westfield State College	No	3100	2150	825	3	20	3234	941
Westmont College	No	950	713	351	42	72	1276	9
Winona State University	No	3325	2047	1301	20	45	5800	872
Winthrop University	No	2320	1805	769	24	61	3395	670
Worcester State College	No	2197	1515	543	4	26	3089	2029
Yale University	Yes	10705	2453	1317	95	99	5217	83

In [36]:
```
1 df.index
```

Out[36]:
```
Index(['Abilene Christian University', 'Adelphi University', 'Adrian C
ollege',
       'Agnes Scott College', 'Alaska Pacific University', 'Albertson
College',
       'Albertus Magnus College', 'Albion College', 'Albright College'
,
       'Alderson-Broaddus College',
       ...
       'Winthrop University', 'Wisconsin Lutheran College',
       'Wittenberg University', 'Wofford College',
       'Worcester Polytechnic Institute', 'Worcester State College',
       'Xavier University', 'Xavier University of Louisiana',
       'Yale University', 'York College of Pennsylvania'],
      dtype='object', length=777)
```

In [41]:
```
1 print(df[df['Cluster'] != df['Predictions']][['Cluster', 'Predictions
```

	Cluster	Predictions
Angelo State University	0	1
Antioch University	1	0
Arkansas Tech University	0	1
Baylor University	1	0
Bemidji State University	0	1
Bloomsburg Univ. of Pennsylvania	0	1
Boston University	1	0
Brigham Young University at Provo	1	0
Brown University	1	0
Carnegie Mellon University	1	0
Castleton State College	0	1
Central Connecticut State University	0	1
Central Missouri State University	0	1
Central Washington University	0	1
Christopher Newport University	0	1
Clinch Valley Coll. of the Univ. of Virginia	0	1
College of Charleston	0	1
College of William and Mary	0	1
Columbia University	1	0
Dartmouth College	1	0
Delta State University	0	1
Dickinson State University	0	1
Duke University	1	0
East Tennessee State University	0	1
Eastern Connecticut State University	0	1
Eastern Illinois University	0	1
Emory University	1	0
Emporia State University	0	1

Evergreen State College	0	1
Fayetteville State University	0	1
...	...	...
University of Southern California	1	0
University of Southern Colorado	0	1
University of Southern Indiana	0	1
University of Southern Mississippi	0	1
University of Texas at Arlington	0	1
University of Texas at San Antonio	0	1
University of West Florida	0	1
University of Wisconsin-Stout	0	1
University of Wisconsin-Superior	0	1
University of Wisconsin-Whitewater	0	1
University of Wisconsin at Green Bay	0	1
University of Wyoming	0	1
Valley City State University	0	1
Vanderbilt University	1	0
Villanova University	1	0
Virginia State University	0	1
Wake Forest University	1	0
Washington University	1	0
Wayne State College	0	1
West Chester University of Penn.	0	1
West Liberty State College	0	1
Western Carolina University	0	1
Western State College of Colorado	0	1
Western Washington University	0	1
Westfield State College	0	1
Westmont College	0	1
Winona State University	0	1
Winthrop University	0	1
Worcester State College	0	1
Yale University	1	0

[172 rows x 2 columns]