lesson_9

June 26, 2016

In [1]:
```
from k_means_cluster import *
```

In [2]:
```
from sklearn.preprocessing import MinMaxScaler
```

0.0.1 Sidenotes

In [3]:
```
#convert to and from numpy float data type
y = numpy.float32(1.0)
type(numpy.zeros(1,y).tolist()[0])
```

Out[3]:
```
float
```

In [4]:
```
#sidenote# return abritrary set of values in dict without popping
next(data_dict.itervalues())
```

Out[4]:
```
{'bonus': 600000,
 'deferral_payments': 'NaN',
 'deferred_income': 'NaN',
 'director_fees': 'NaN',
 'email_address': 'mark.metts@enron.com',
 'exercised_stock_options': 'NaN',
 'expenses': 94299,
 'from_messages': 29,
 'from_poi_to_this_person': 38,
 'from_this_person_to_poi': 1,
 'loan_advances': 'NaN',
 'long_term_incentive': 'NaN',
 'other': 1740,
 'poi': False,
 'restricted_stock': 585062,
 'restricted_stock_deferred': 'NaN',
 'salary': 365788,
 'shared_receipt_with_poi': 702,
 'to_messages': 807,
 'total_payments': 1061827,
 'total_stock_value': 585062}
```

0.0.2 Quiz on Computing Rescaled Features

Apply feature scaling to your k-means clustering code from the last lesson, on the "salary" and "exercised_stock_options" features (use only these two features). What would be the rescaled value of a "salary" feature that had an original value of $200,000, and an "exercised_stock_options" feature of $1 million? (Be sure to represent these numbers as floats, not integers!)

In [5]:
```
min_max_scaler = MinMaxScaler()
```

In [6]:
```
min_max_scaler.fit(data)
```

Out[6]:
```
MinMaxScaler(copy=True, feature_range=(0, 1))
```

In [7]:
```
# salary of £200k, stock options of £1m put into array of correct format
quiz_data_to_scale = numpy.array([0, 200000, 1000000, 0]).reshape(1,-1)
min_max_scaler.transform(quiz_data_to_scale)  #correct output
```

Out[7]:
```
array([[ 0.        ,  0.17997621,  0.02911345,  0.        ]])
```

In [8]:
```
scaled_data = min_max_scaler.fit_transform(data)
```

First attempt

In [9]:
```
# make dict mapping salary data to scaled salary data
# tried this method thinking quiz data was in dataset (it is not)
scaled_salary_dict = {i : s for i, s in
                      zip(data[:,1], scaled_data[:,1])}

print scaled_salary_dict.get(numpy.float32(278601))

print scaled_salary_dict.keys()[1] # can't remember relevance of this
```

```
0.250707756435
216582.0
```

In [ ]: