kmeans

October 30, 2019

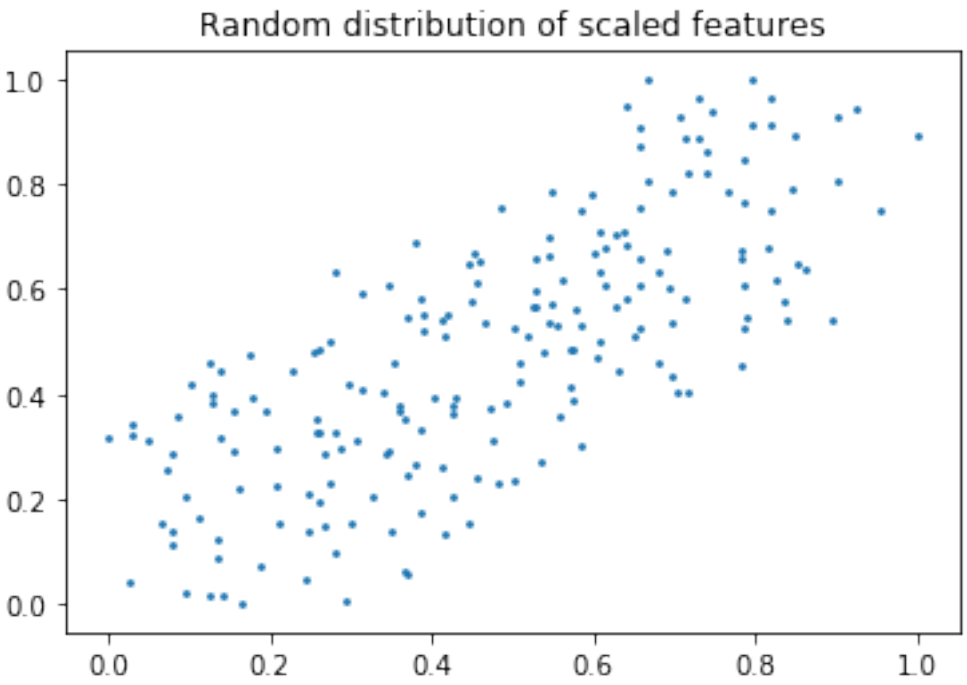
[1]:
```
import pandas as pd
import numpy as np
from sklearn import cluster
from sklearn import preprocessing
from sklearn import metrics
from matplotlib import pyplot as plt
```

[34]:
```
import random
data = []
for x in range(200):
    data.append({'x': x + random.randint(0, 100), 'y': x + random.randint(0,␣
 ↪100)})
df = pd.DataFrame(data)
```

[35]:
```
scaler = preprocessing.MinMaxScaler()
scaler.fit(df[['x']])
df.x = scaler.transform(df[['x']])
scaler.fit(df[['y']])
df.y = scaler.transform(df[['y']])
```

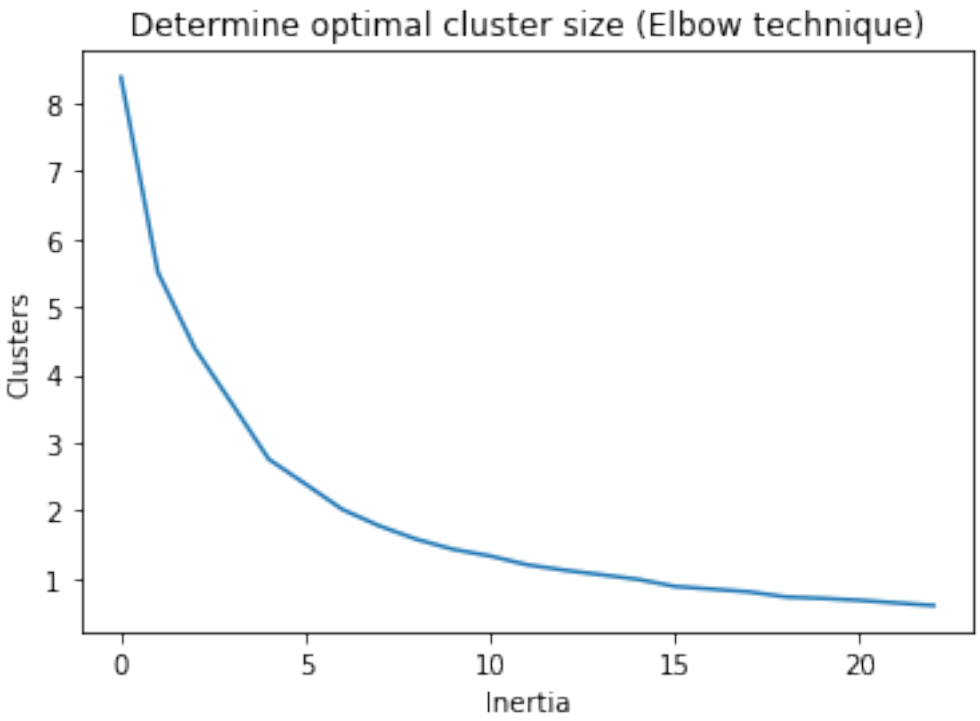
[54]:
```
plt.title('Random distribution of scaled features')
plt.scatter(df.x, df.y, s=3)
```

[54]:
```
<matplotlib.collections.PathCollection at 0x7fc0a2004518>
```



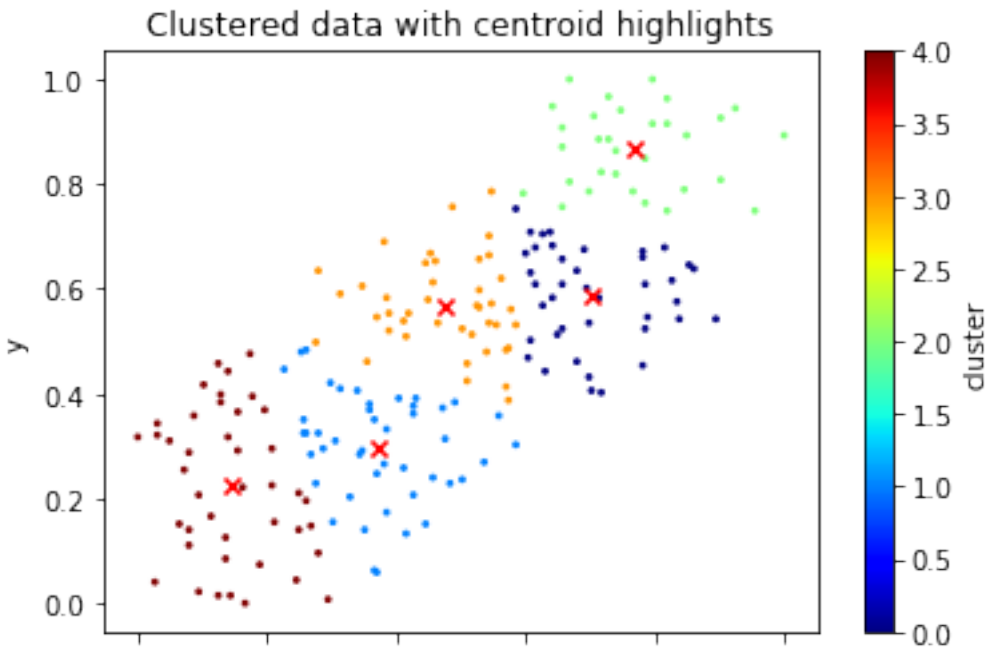
[49]:
```
test = []
for t in range(2, 25):
    km = cluster.KMeans(n_clusters=t)
    km.fit(df[['x', 'y']])
    test.append(km.inertia_)
plt.title('Determine optimal cluster size (Elbow technique)')
plt.xlabel('Inertia')
plt.ylabel('Clusters')
plt.plot(test)
```

[49]:
```
[<matplotlib.lines.Line2D at 0x7fc0a24687f0>]
```



[53]:
```
km = cluster.KMeans(n_clusters=5)
km.fit(df[['x', 'y']])
df['cluster'] = km.labels_
centroids = km.cluster_centers_
q = df.plot.scatter(x='x', y='y', c='cluster', colormap='jet', s=3)
q.set_title('Clustered data with centroid highlights')
q.scatter(centroids[:,0], centroids[:,1], marker='x', color='red')
```

[53]:
```
<matplotlib.collections.PathCollection at 0x7fc0a212ac88>
```


Clustered data with centroid highlights
y
1.0
0.8
0.6
0.4
0.2
0.0
duster
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0