การจัดกลุ่มข้อมูล (Clustering/Segmentation)

Comparing different clustering algorithms on toy datasets การใช้ Classification แม่นกว่าแน่นอน เพราะเป็น supervisation

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import  KMeans
import seaborn as sns
%matplotlib inline
```

```
# โหลดข้อมูล
telco_user=pd.read_csv(
    "https://github.com/praisan/hello-world/raw/master/churn_data.csv")
telco_user.columns
```

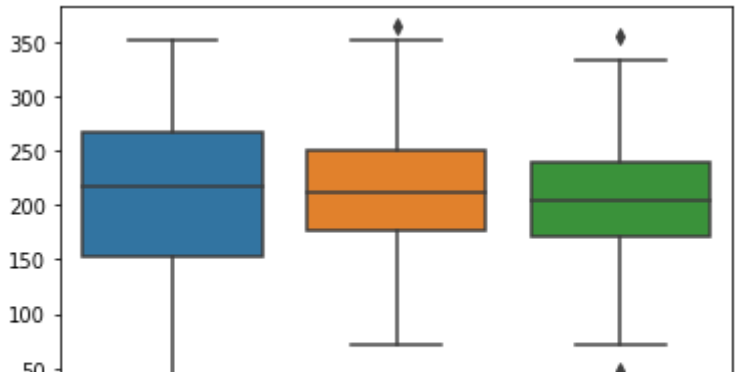
```
Index(['AccountLength', 'VMailMessage', 'DayMins', 'EveMins', 'NightMins',
       'IntlMins', 'CustServCalls', 'IntPlan', 'VMailPlan', 'DayCalls',
       'DayCharge', 'EveCalls', 'EveCharge', 'NightCalls', 'NightCharge',
       'IntlCalls', 'IntlCharge', 'State', 'AreaCode', 'Phone', 'Churn'],
      dtype='object')
```

```
# เลือกข้อมูลที่ต้องการนำไปวิเคราะห์ ขึ้นอยู่กับ Business ไม่ใช่ algor
user_feature=telco_user.loc[telco_user.Churn==1,
                            ['DayMins', 'EveMins','NightMins']]
user_feature.describe()
```

	DayMins	EveMins	NightMins
count	483.000000	483.000000	483.000000
mean	206.914079	212.410145	205.231677
std	68.997792	51.728910	47.132825
min	0.000000	70.900000	47.400000
25%	153.250000	177.100000	171.250000
50%	217.600000	211.300000	204.800000
75%	265.950000	249.450000	239.850000
max	350.800000	363.700000	354.900000

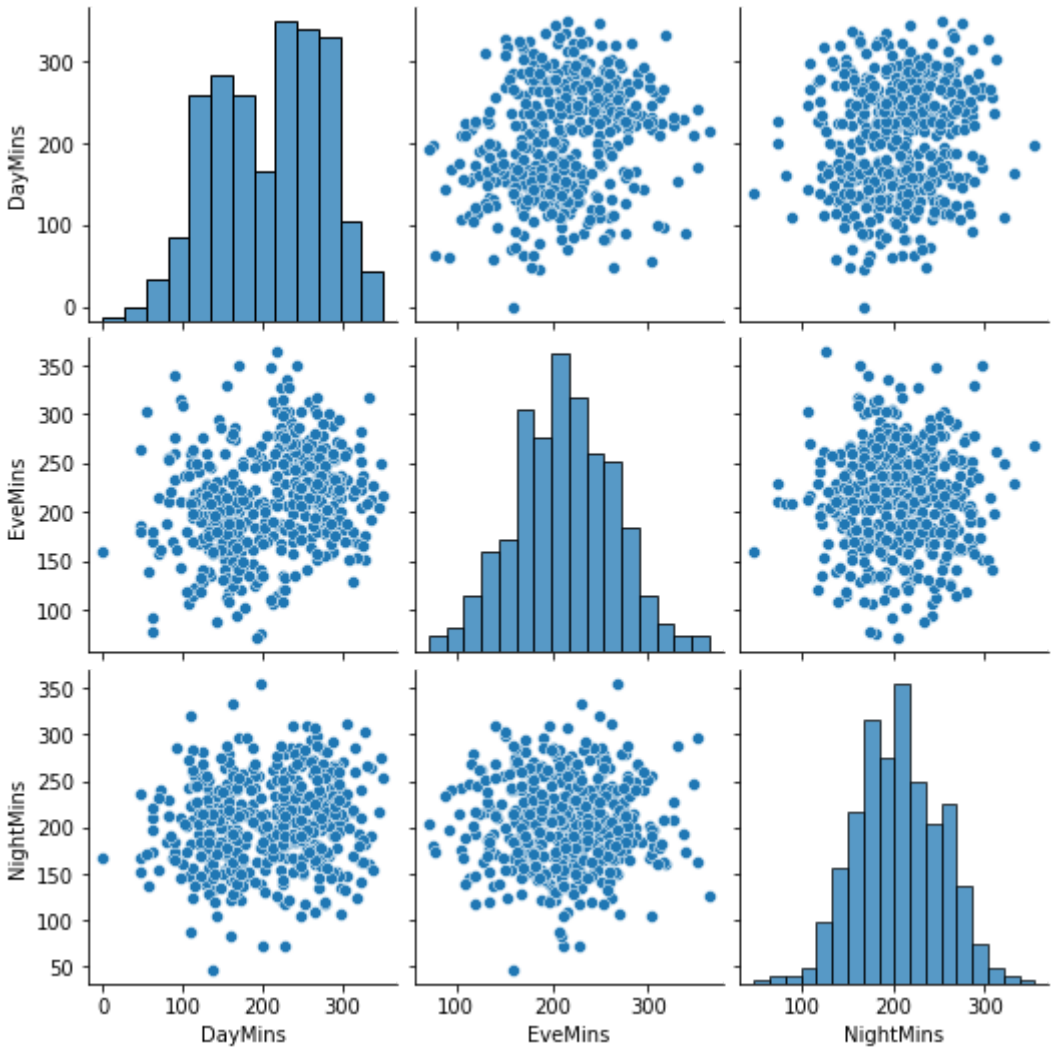
```
sns.boxplot(data=user_feature)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b880af1d0>
```




```
# plot ดูการกระจายตัวของข้อมูลและ plot แต่ละคู่ของ feature //เห็นว่าแยกเป็นสองกลุ่มได้ดี
sns.pairplot(user_feature)
```

```
<seaborn.axisgrid.PairGrid at 0x7f6b880af080>
```

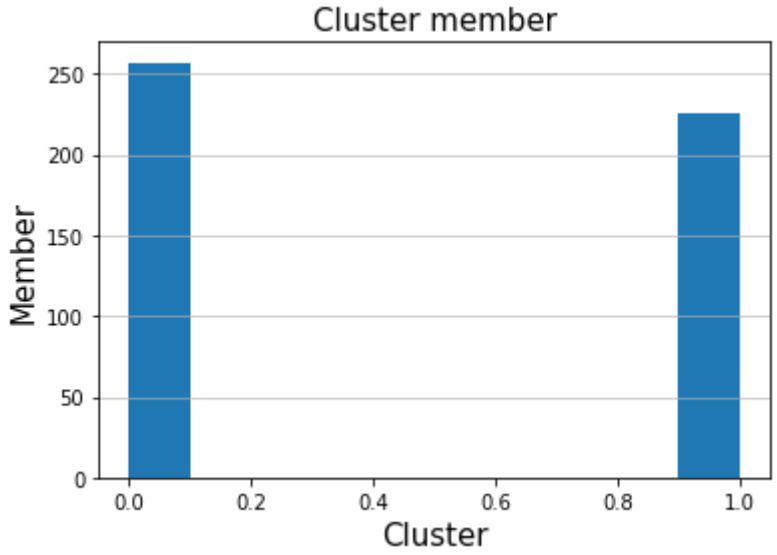



```
# จัดกลุ่มข้อมูลโดยใช้จำนวน Cluster เป็น 2 คือการจัดกลุ่ม
kmean=KMeans(n_clusters=2)
user_segment_2c = kmean.fit(user_feature)


# แสดงจำนวนสมาชิกใน Cluster
# labels_ เป็น attribute เพื่อแบ่งกลุ่ม
# มี _ คือ attribute ไม่มีคือ method
plt.hist(user_segment_2c.labels_)
plt.grid(axis='y', alpha=0.75)
```

```
plt.xlabel('Cluster',fontsize=15)
plt.ylabel('Member',fontsize=15)
plt.title('Cluster member',fontsize=15)
```

```
Text(0.5, 1.0, 'Cluster member')
```




```
# ผลที่ได้จากการจัดกลุ่มข้อมูล
print("- Cluster center")
print(user_segment_2c.cluster_centers_)
print("- WCSSE")
print(user_segment_2c.inertia_)
print("- Cluster/Segment")
print(user_segment_2c.labels_)
```

```
- Cluster center
[[261.60272374 234.18871595 213.70233463]
 [144.72389381 187.64424779 195.59911504]]
- WCSSE
2712540.332287456
- Cluster/Segment
[1 0 1 0 1 1 1 1 1 0 1 0 1 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 1 0 1 1 1 0 1
 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 1 0 1 0 1 1 0 0 1 0 1 0
 0 1 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 1 0 1 1 0 1 1 0 0 1 0 1 0 1 1 0
 0 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 0 0 0 1 1 0 1 0 1 1 0 1 0 1 1 1 0 0
 1 0 0 0 1 1 1 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1 0 0 1 0 1 1 0 0 0 1 0 0 1 1 1 1 1
 1 1 0 1 1 0 0 0 1 1 0 0 1 1 1 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0 1 0 1 0 1 1 0
 0 0 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 1 1
 1 1 0 0 1 1 1 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 0 0 0 1 1
 0 1 0 1 1 0 0 0 1 1 0 0 0 1 0 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 0 0 1 1 1
 0 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 1 0 0 0
 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0 0 1 0 1 1 0 1
 1 0 1 0 0 1 1 0 0 1 0 1 0 0 0 1 0 1 1 1 0 0 0 1 1 0 1 1 0 0 0 1 1 0 1 0 1
 0 1 0 1 0 1 1 0 1 0 0 1 0 0 1 0 1 0 0 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 1 1
 0 1]
```

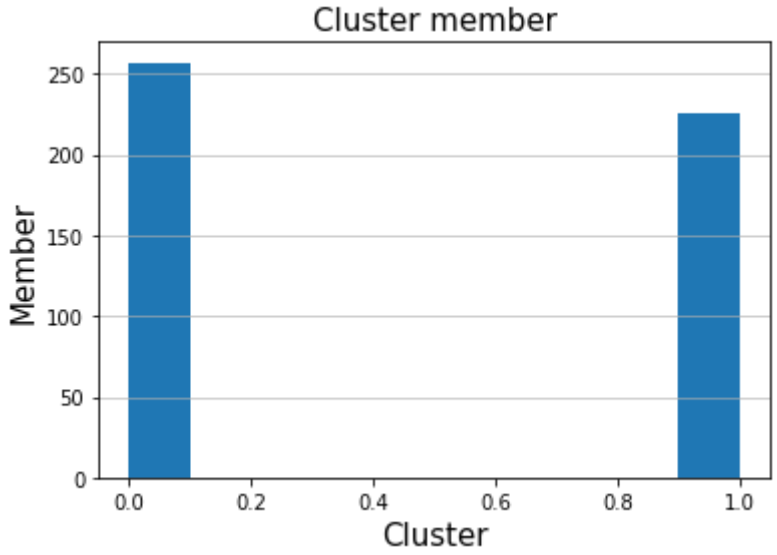
```
# ทดลองบันทึก Model ของการจัดกลุ่ม
from sklearn.externals import joblib
joblib.dump(user_segment_2c,'finalized_model.model')
#pickle.dumps(user_segment_2c, 'telco_segment_2c.model')
```

```
['finalized_model.model']
```

```
# ทดลองโหลด โมเดลมาใช้
segment_model=joblib.load('finalized_model.model')
segment_result=segment_model.predict(user_feature)

plt.hist(segment_result)
plt.grid(axis='y', alpha=0.75)
plt.xlabel('Cluster',fontsize=15)
plt.ylabel('Member',fontsize=15)
plt.title('Cluster member',fontsize=15)
```

```
Text(0.5, 1.0, 'Cluster member')
```



ฝึกปฏิบัติ 1

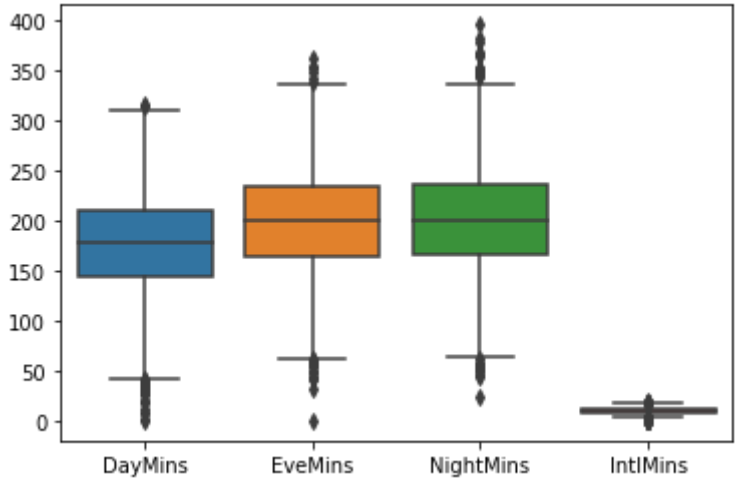
ใช้ fearure DayMins, EveMins, NightMins, IntlMins มาจัดกลุ่มผู้ใช้ออกเป็น 3 กลุ่ม

```
# เลือกข้อมูลที่ต้องการนำไปวิเคราะห์
user_4_features=telco_user.loc[telco_user.Churn==0,['DayMins',  'EveMins',
                                                    'NightMins','IntlMins']]
user_4_features.head()
```

	DayMins	EveMins	NightMins	IntlMins
0	265.1	197.4	244.7	10.0
1	161.6	195.5	254.4	13.7
2	243.4	121.2	162.6	12.2
3	299.4	61.9	196.9	6.6
4	166.7	148.3	186.9	10.1

```
sns.boxplot(data=user_4_features)
```

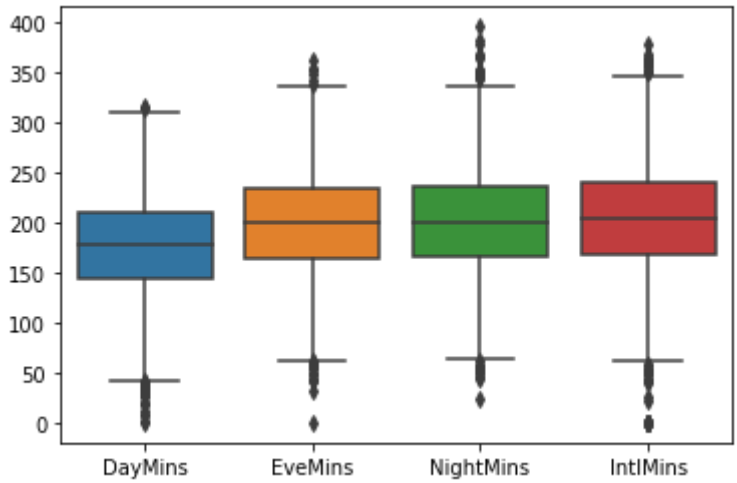
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b87aeee80>
```




```
#ต้องการเพิ่มบทบาทให้ IntlMins (เนื่องจากหน่วยมันเท่ากันอยู่แล้ว)
# หรือจะทำ standardization ก็ได้ ให้ sd เท่ากัน mean เท่ากัน คือใช้แก้ปัญหาหน่วยไม่เท่ากัน
user_4_features['IntlMins']=user_4_features['IntlMins']*20


sns.boxplot(data=user_4_features)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b8864a668>
```

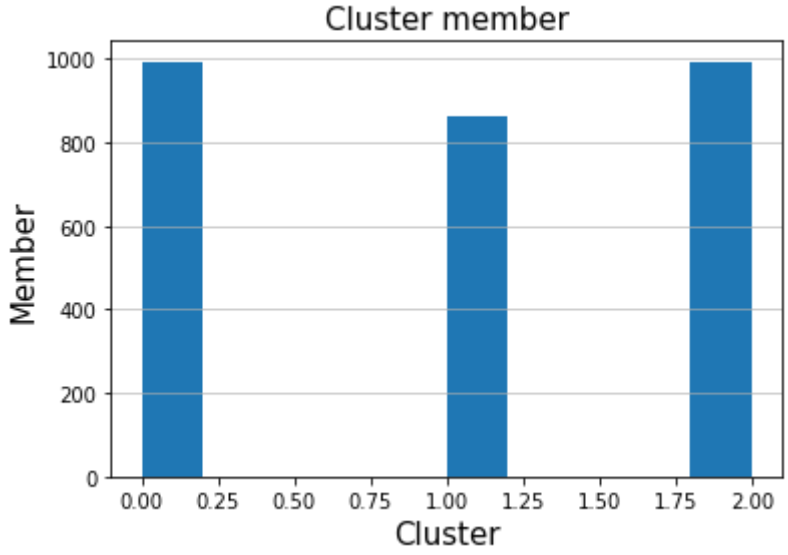



```
# จัดกลุ่มข้อมูลโดยใช้จำนวน Cluster เป็น 3
user_segment_3c4f = kmean.set_params(n_clusters=3).fit(user_4_features)


# ทดลองบันทึก Model ของการจัดกลุ่ม
joblib.dump(user_segment_3c4f,'finalized_model_3c4f.model')
#ทดลองโหลดโมเดลมาใช้
segment_model=joblib.load('finalized_model_3c4f.model')
segment_result=segment_model.predict(user_4_features)


# แสดงจำนวนสมาชิกใน Cluster
plt.hist(segment_result)
plt.grid(axis='y', alpha=0.75)
plt.xlabel('Cluster',fontsize=15)
plt.ylabel('Member',fontsize=15)
plt.title('Cluster member',fontsize=15)
```

```
Text(0.5, 1.0, 'Cluster member')
```



อธิบายเพิ่มเติม

- kmean=KMeans() คือการสร้าง Object ชนิด KMeans ซึ่งการตั้งค่าต่าง ๆ สามารถกำหนดทีหลังได้ โดยใช้ฟังก์ชัน set_params

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

- result=kmean.set_params(n_clusters=n_cluster).fit(data) คือการกำหนดจำนวน cluster ตามที่เราต้องการแล้วทำการจัดกลุ่มข้อมูลที่อยู่ใน data จากนั้นเก็บผลของการจัดกลุ่มไว้ใน result

```
คำสั่งที่ใช้บ่อย
- fit(data) จัดกลุ่มข้อมูลที่ระบุเพื่อให้ได้โมเดล ผลลัพธ์ที่ใช้บ่อยคือ
   - cluster_centers_ เก็บ centroids หลังจากเรียนรู้จากข้อมูลแล้ว
   - inertia_ เก็บ WCSSE
   - labels_ เก็บ label ระบุว่าข้อมูลใน data นั้นอยู่กลุ่มไหน
- transform(data) วัดระยะห่างระหว่างข้อมูลใน data กับ centroid ทุกตัว
- predict(data) ระบุกลุ่มให้ข้อมูลใน data
```

สิ่งที่ควรวิเคราะห์เพิ่มเติม

- การทำ standardization ก่อนการจัดกลุ่มข้อมูล
- การใช้ Clustering แบบอื่นเพื่อจัดกลุ่มที่แต่ละกลุ่มมีขนาดและรูปร่างแตกต่างกันได้ดี เช่น Gaussian Mixture Model (GMM)

หาจำนวน Cluster ที่เหมาะสม (เลือกผลการจัดกลุ่มที่เหมาะสม)

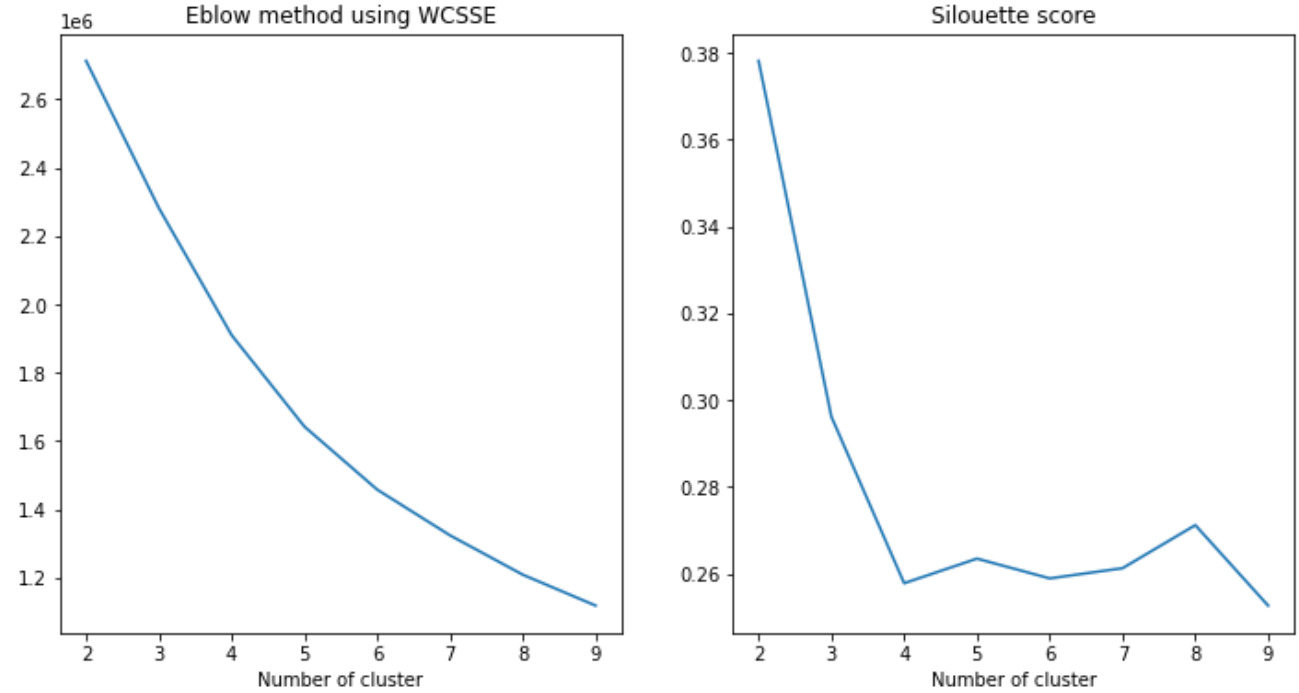
- ใช้วิธีดูจุดหักศอก (Elbow method)

- Silhouette

```
from sklearn.metrics import silhouette_score
#กำหนดช่วง 2-9
clusters=range(2,10)
#วน loop จัดกลุ่มตามจน. cluster เพื่อคำนวณ silhouette score
n_clus=len(clusters)
wcsse=[0]*n_clus
silhouette_avg=[0]*n_clus
for i in range(0,n_clus):
    user_segment = kmean.set_params(n_clusters=clusters[i]).fit(user_feature)
    silhouette_avg[i] = silhouette_score(user_feature, user_segment.labels_)
    wcsse[i]=user_segment.inertia_


#จะเห็นว่าศอกมนไม่ค่อยหัก เลยใช้ silhouettว่าย จึงเห็นว่าควรใช้2 >3 >8
f, axes = plt.subplots(1,2)
f.set_size_inches(w=12,h=6)
sns.lineplot(x=clusters,y=wcsse,ax=axes[0])
axes[0].set_title("Eblow method using WCSSE")
axes[0].set_xlabel('Number of cluster')
sns.lineplot(x=clusters,y=silhouette_avg,ax=axes[1])
axes[1].set_title("Silouette score")
axes[1].set_xlabel('Number of cluster')
```

```
Text(0.5, 0, 'Number of cluster')
```



Application

- Segmentation

- Outlier/Anomaly analysis

Segmentation

- ตีความแต่ละ cluster
- บันทึกโมเดลเพื่อใช้งานกับข้อมูลใหม่
- ระบุ segment/cluster ของข้อมูลใหม่

```
selected_n_cluster=3
user_segment = kmean.set_params(n_clusters=
                                selected_n_cluster).fit(user_4_features)
centroids=pd.DataFrame(user_segment.cluster_centers_,
                       columns=['DayMins',  'EveMins',  'NightMins','IntlMins'])
centroids
```

	DayMins	EveMins	NightMins	IntlMins
0	139.720829	211.906673	215.310313	239.470172
1	175.634028	207.118287	217.599653	144.020833
2	209.949047	179.285356	169.941424	218.441324

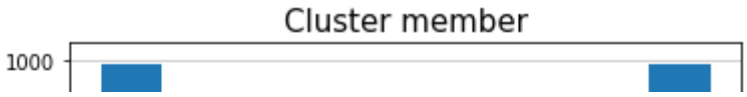
```
# บันทึกโมเดลไว้ใช้ต่อ
from sklearn.externals import joblib
joblib.dump(user_segment,'finalized_model.model')
```

```
['finalized_model.model']
```

```
# ทดลองโหลด โมเดลมาใช้
segment_model=joblib.load('finalized_model.model')
segment_result=segment_model.predict(user_4_features)

plt.hist(segment_result)
plt.grid(axis='y', alpha=0.75)
plt.xlabel('Cluster',fontsize=15)
plt.ylabel('Member',fontsize=15)
plt.title('Cluster member',fontsize=15)
```

```
Text(0.5, 1.0, 'Cluster member')
```



Final Model pipeline


```
from sklearn.pipeline import Pipeline
from sklearn.base import TransformerMixin
from sklearn.preprocessing import StandardScaler
# class SelectFeature Extend มาจาก class TransformerMixin
class SelectFeature(TransformerMixin):
    def __init__(self,feature=['DayMins',   'EveMins','NightMins']):# constructure
        """Select columns.

        """
        self.feature=feature

    def fit(self, X, y=None):# function
        return self

    def transform(self, X, y=None):# function
        return X[self.feature]
class ScaleFeature(TransformerMixin): #หรือจะใช้ z-score เพื่อทำ normalization ก็ได้
    def fit(self, X, y=None):
        return self

    def transform(self, X, y=None):
        X['IntlMins']=X['IntlMins']*20
        return X

selectFeature=SelectFeature(feature=['DayMins', 'EveMins','NightMins'])
#normalize_zscore=StandardScaler(with_mean=True, with_std=True)
kmean=KMeans(n_clusters=8)

models = Pipeline([
  ('feature', selectFeature),
 # ('zscore', normalize_zscore),
  ('segmentation', kmean)
])

model=models.fit(telco_user.loc[telco_user.Churn==0]) # ฝึกโมเดลใน Pipeline

from sklearn.externals import joblib
joblib.dump(model,'pipeline_model.model')
```

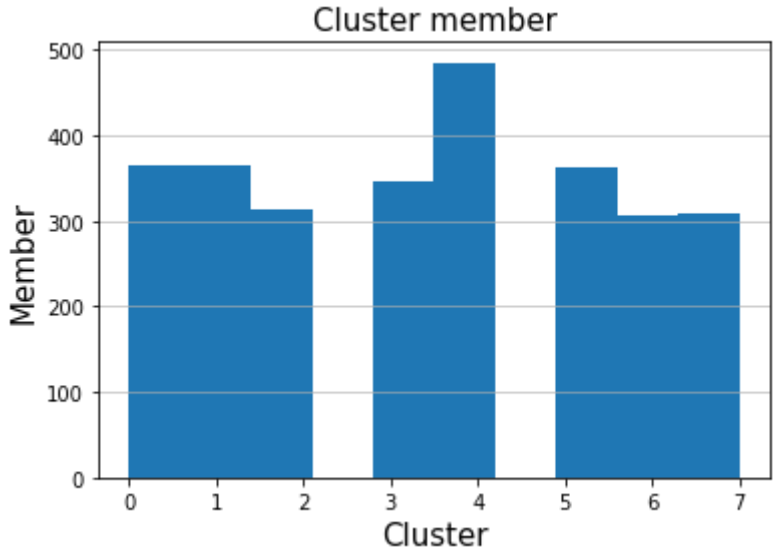
```
['pipeline_model.model']
```

```
segment_model=joblib.load('pipeline_model.model')
segment_result=segment_model.predict(telco_user.loc[telco_user.Churn==0])

plt.hist(segment_result)
plt.grid(axis='y', alpha=0.75)
```

```
plt.xlabel('Cluster',fontsize=15)
plt.ylabel('Member',fontsize=15)
plt.title('Cluster member',fontsize=15)
```

```
Text(0.5, 1.0, 'Cluster member')
```

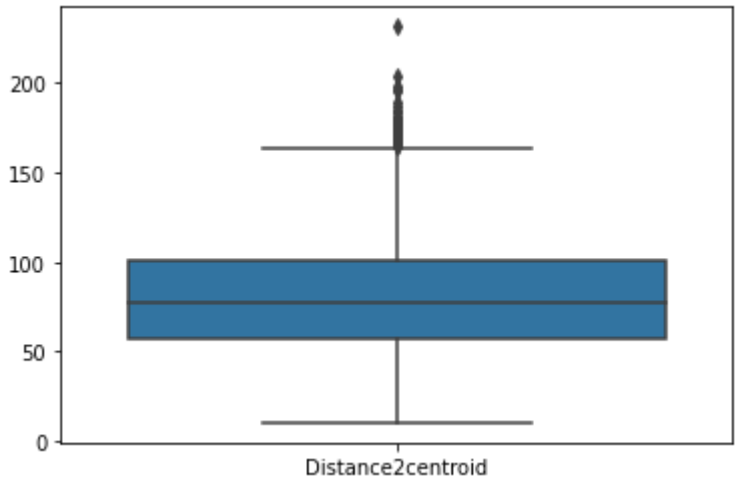


Outlier/Anomaly analysis

```
# หาระยะระหว่างข้อมูลกับจุด
#distance to centroids
distance2centroids =np.array(user_segment.transform(user_4_features))
distance2my_centroid=pd.DataFrame(distance2centroids.min(axis=1),
                                  columns=['Distance2centroid'])
```

```
sns.boxplot(data=distance2my_centroid)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b8956e668>
```

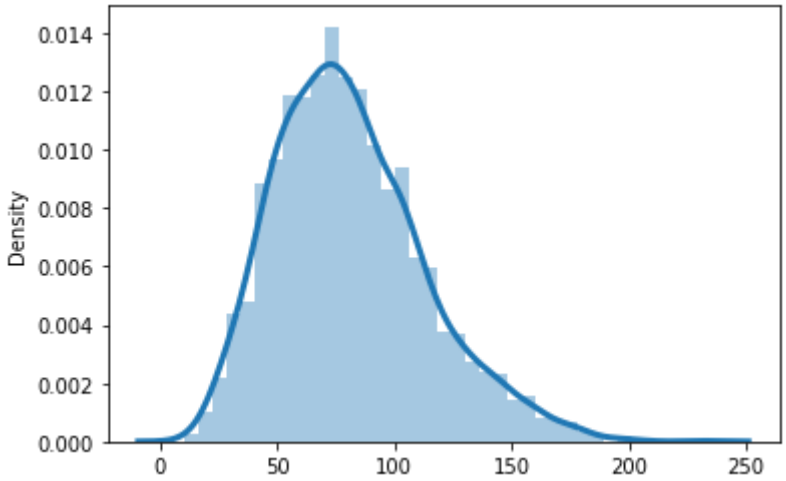



```
distance2my_centroid.describe()
```

	Distance2centroid
count	2850.000000
mean	81.056253
std	31.986094
min	10.332170
25%	57.571716
50%	77.529908
75%	100.506335
max	231.352399

```
sns.distplot(distance2my_centroid, hist = True, kde = True,
             kde_kws = {'linewidth': 3}) #แสดงข้อมูลแบบ histogram
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/distributions.py:2551: FutureW
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b89625e10>
```




```
# ระบุแถวที่เป็น outlier ที่อยู่ห่างจาก centroid ของตัวเองมากกว่า Q3 + 1.5 * IQR
# ระบุได้ด้วยการใช้มิติเดียว ด้วย distance to centroid ด้วยวิธี IQR
Q1 = distance2my_centroid.quantile(0.25)
Q3 = distance2my_centroid.quantile(0.75)
IQR = Q3 - Q1

outlier_index=(distance2my_centroid > (Q3 + 1.5 * IQR))
outlier_index.any(axis=1).value_counts().plot.bar()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b878a6e80>
```


```
cleaned_data=user_4_features.loc[list(~outlier_index.Distance2centroid)]
cleaned_data.head() # เลือกเอาเฉพาะข้อมูลที่ไม่เป็น outlier
```

	DayMins	EveMins	NightMins	IntlMins
0	265.1	197.4	244.7	200.0
1	161.6	195.5	254.4	274.0
2	243.4	121.2	162.6	244.0
4	166.7	148.3	186.9	202.0
5	223.4	220.6	203.9	126.0

```
outlier_data=user_4_features.loc[list(outlier_index.Distance2centroid)]
outlier_data.head() # เลือกเอาเฉพาะข้อมูลที่เป็น outlier
```

	DayMins	EveMins	NightMins	IntlMins
3	299.4	61.9	196.9	132.0
9	258.6	222.0	326.4	224.0
179	232.1	292.3	201.2	0.0
268	94.4	136.2	147.4	90.0
315	60.4	306.2	123.9	248.0

ฝึกปฏิบัติ 2

ใช้ fearure DayMins, EveMins, NightMins, IntlMins , VMailMessage

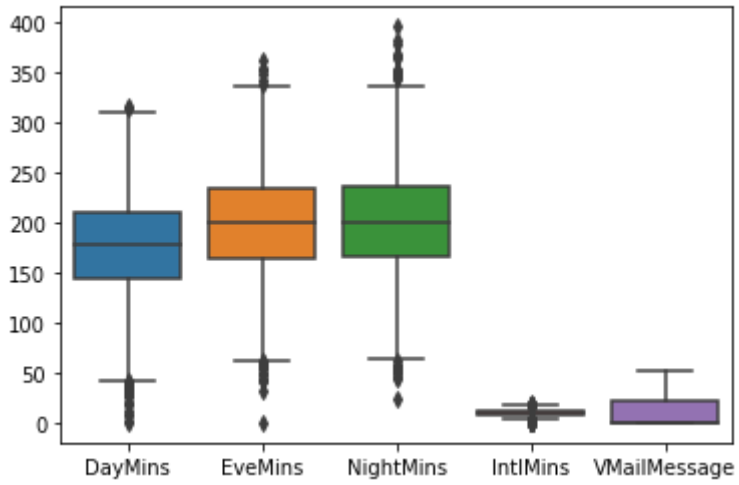
1. ทดลองสร้างโมเดลโดยหาว่าควรใช้จำนวนกลุ่มเท่าไหร่จึงจะเหมาะสมแล้วบันทึกไว้ใช้งาน
2. หาผู้ใช้ที่มีพฤติกรรมสุดโต่งจากกลุ่มของตัวเอง (outlier)

```
user_5_features=telco_user.loc[telco_user.Churn==0,['DayMins',  'EveMins',
                               'NightMins','IntlMins','VMailMessage']]
user_5_features.head()
```

	DayMins	EveMins	NightMins	IntlMins	VMailMessage
0	265.1	197.4	244.7	10.0	25
1	161.6	195.5	254.4	13.7	26
2	243.4	121.2	162.6	12.2	0

```
sns.boxplot(data=user_5_features)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b89337b38>
```

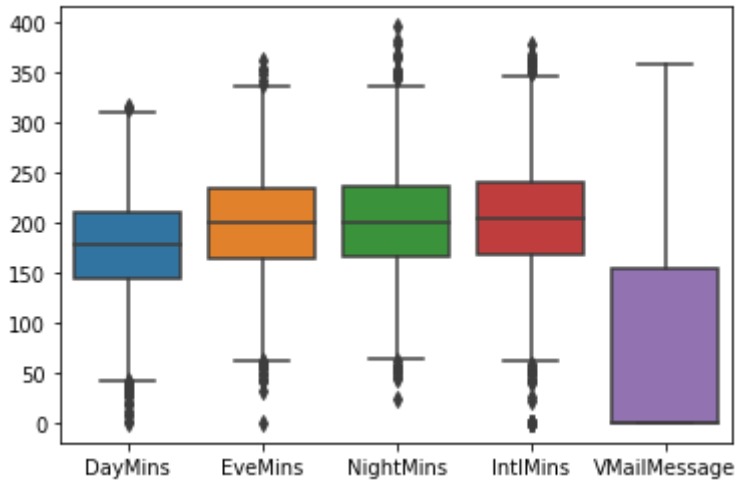



```
# ดูการกระจายของข้อมูล
sns.pairplot(user_5_features)
```

```
# เพิ่มขนาดสักหน่อย
user_5_features['IntlMins']=user_5_features['IntlMins']*20
user_5_features['VMailMessage']=user_5_features['VMailMessage']*7
```

```
# ได้ไซส์ใหม่แล้วว
sns.boxplot(data=user_5_features)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b8881f0f0>
```

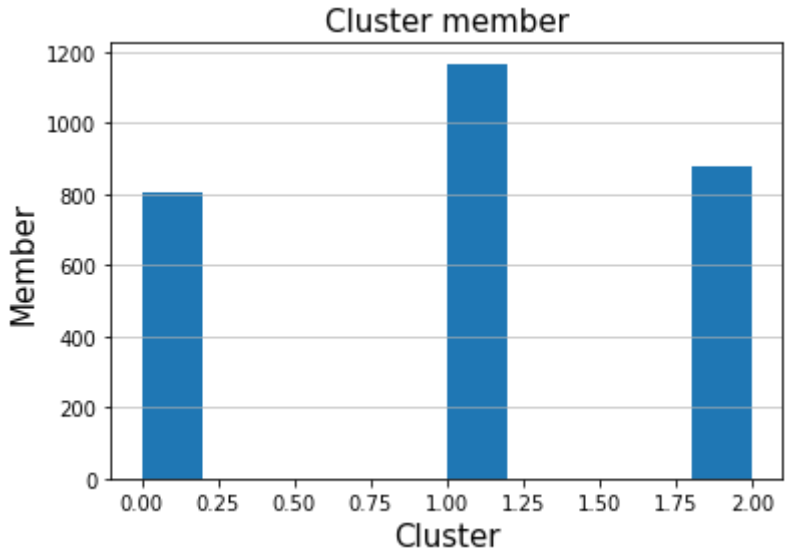



```
# จัดกลุ่มข้อมูลโดยใช้จำนวน cluster เป็น 3
user_segment_3c5f = kmean.set_params(n_clusters=3).fit(user_5_features)
```

```
# บันทึก model เก็บไว้ใช้
joblib.dump(user_segment_3c5f,'finalized_model_3c5f.model')
segment_model=joblib.load('finalized_model_3c5f.model')
segment_result=segment_model.predict(user_5_features)


# แสดงจำนวนสมาชิกใน cluster
plt.hist(segment_result)
plt.grid(axis='y', alpha=0.75)
plt.xlabel('Cluster',fontsize=15)
plt.ylabel('Member',fontsize=15)
plt.title('Cluster member',fontsize=15)
```

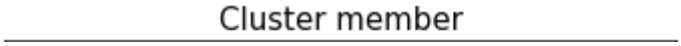
```
Text(0.5, 1.0, 'Cluster member')
```




```
# ทดลองโหลด โมเดลที่สร้างเองมาใช้ เย่ๆ
segment_model=joblib.load('finalized_model_3c5f.model')
segment_result=segment_model.predict(user_5_features)

plt.hist(segment_result)
plt.grid(axis='y', alpha=0.75)
plt.xlabel('Cluster',fontsize=15)
plt.ylabel('Member',fontsize=15)
plt.title('Cluster member',fontsize=15)
```

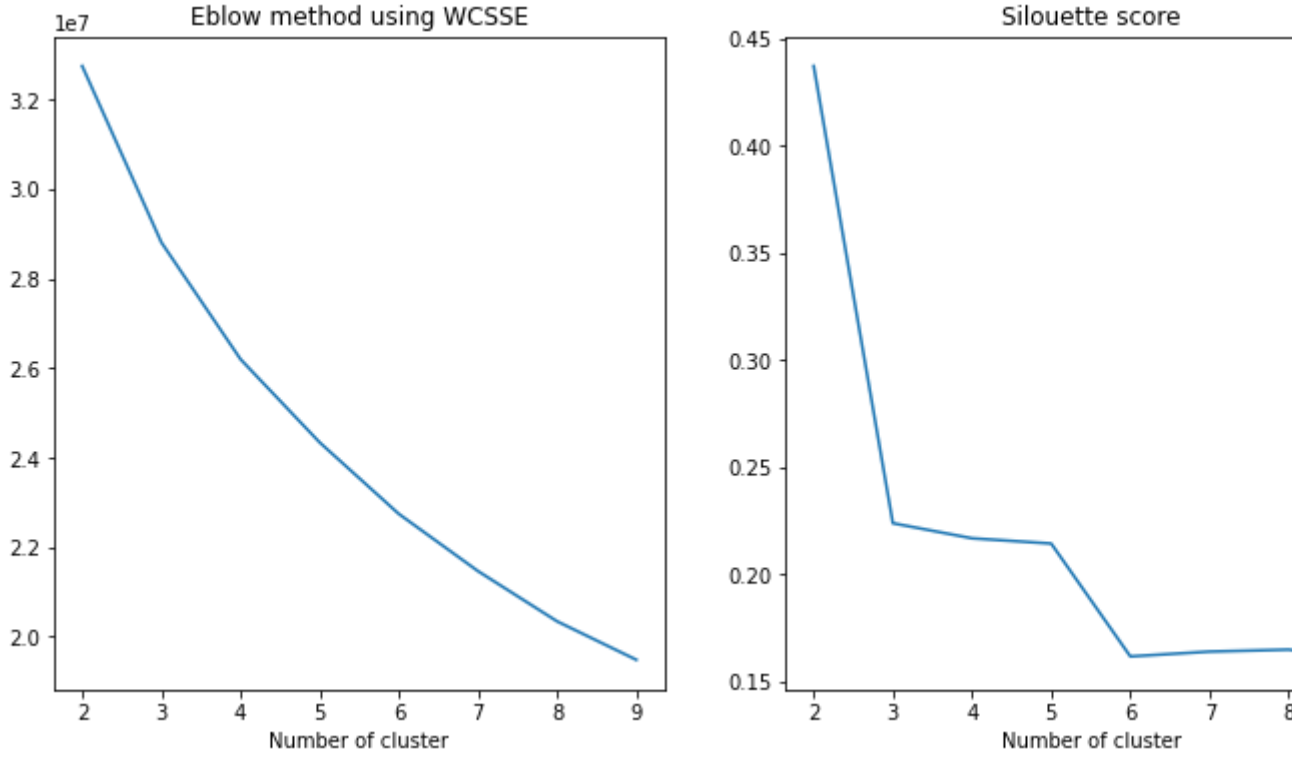
```
Text(0.5, 1.0, 'Cluster member')
```




```
# หาจำนวน cluster ที่เหมาะสมจาก model ของเรา
from sklearn.metrics import silhouette_score
clusters=range(2,10)
n_clus=len(clusters)
wcsse=[0]*n_clus
silhouette_avg=[0]*n_clus
for i in range(0,n_clus):
    user_segment = kmean.set_params(n_clusters=clusters[i]).fit(user_5_features)
    silhouette_avg[i] = silhouette_score(user_5_features, user_segment.labels_)
    wcsse[i]=user_segment.inertia_
```


```
f, axes = plt.subplots(1,2)
f.set_size_inches(w=12,h=6)
sns.lineplot(x=clusters,y=wcsse,ax=axes[0])
axes[0].set_title("Eblow method using WCSSE")
axes[0].set_xlabel('Number of cluster')
sns.lineplot(x=clusters,y=silhouette_avg,ax=axes[1])
axes[1].set_title("Silouette score")
axes[1].set_xlabel('Number of cluster')
```

```
Text(0.5, 0, 'Number of cluster')
```



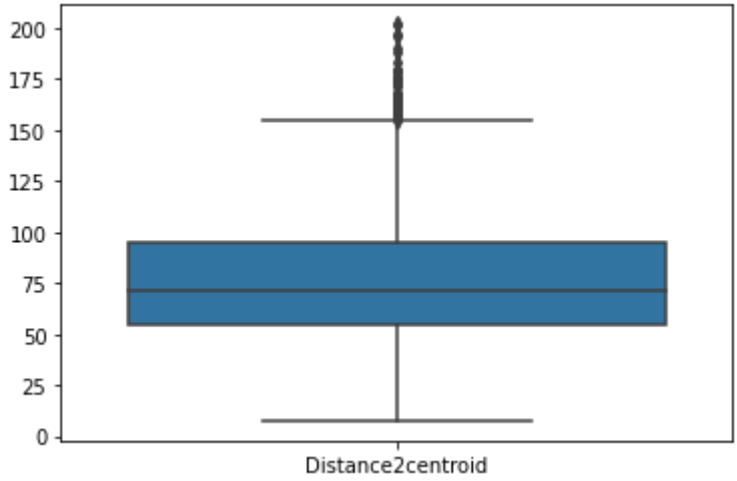

```
# Outlier
distance2centroids =np.array(user_segment.transform(user_5_features))
distance2my_centroid=pd.DataFrame(distance2centroids.min(axis=1),
                                  columns=['Distance2centroid'])


distance2my_centroid.describe()
```

	Distance2centroid
count	2850.000000
mean	76.820765
std	30.572295
min	7.526233
25%	54.788561
50%	71.569340
75%	94.973536
max	201.694821

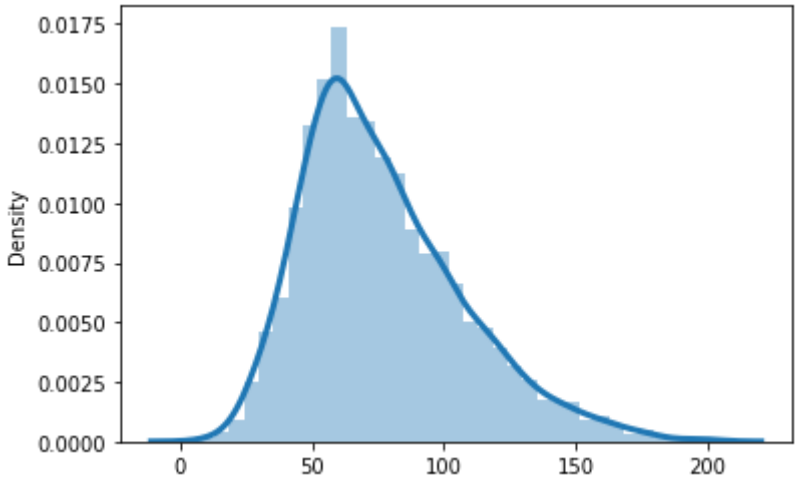
```
sns.boxplot(data=distance2my_centroid)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b88582710>
```




```
sns.distplot(distance2my_centroid, hist = True, kde = True,
                 kde_kws = {'linewidth': 3})
```

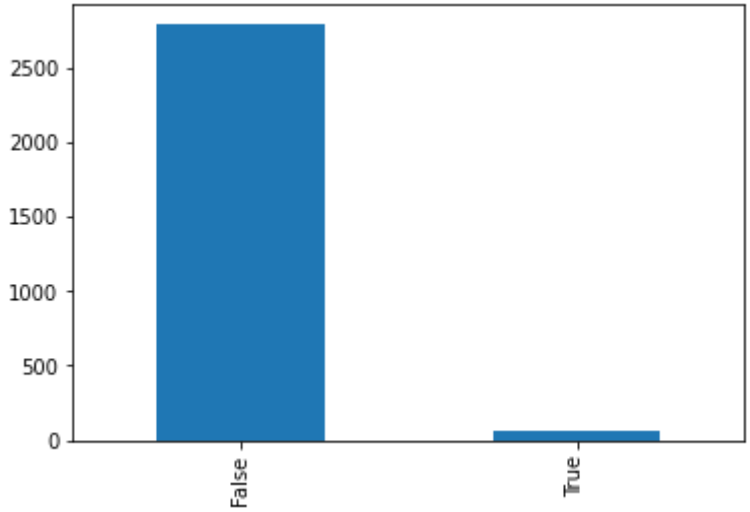
```
/usr/local/lib/python3.6/dist-packages/seaborn/distributions.py:2551: FutureW
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b884d47b8>
```



```
# ระบุแถวที่เป็น outlier ที่อยู่ห่างจาก centroid ของตัวเองมากกว่า Q3 + 1.5 * IQR
Q1 = distance2my_centroid.quantile(0.25)
Q3 = distance2my_centroid.quantile(0.75)
IQR = Q3 - Q1

outlier_index=(distance2my_centroid > (Q3 + 1.5 * IQR))
outlier_index.any(axis=1).value_counts().plot.bar()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b8841d668>
```




```
# เลือกเอาเฉพาะข้อมูลที่ไม่เป็น outlier
cleaned_data=user_5_features.loc[list(~outlier_index.Distance2centroid)]
cleaned_data.head()
```

	DayMins	EveMins	NightMins	IntlMins	VMailMessage
0	265.1	197.4	244.7	10.0	25
1	161.6	195.5	254.4	13.7	26
2	243.4	121.2	162.6	12.2	0
4	166.7	148.3	186.9	10.1	0
5	223.4	220.6	203.9	6.3	0

```
# เลือกเอาเฉพาะข้อมูลที่เป็น outlier
outlier_data=user_5_features.loc[list(outlier_index.Distance2centroid)]
outlier_data.head()
```

	DayMins	EveMins	NightMins	IntlMins	VMailMessage
3	299.4	61.9	196.9	6.6	0
9	258.6	222.0	326.4	11.2	37
179	232.1	292.3	201.2	0.0	0
268	94.4	136.2	147.4	4.5	48
315	60.4	306.2	123.9	12.4	0