


```
In [ ]: plot_model(sc)
```

KMeans Clustering

```
In [ ]: kmeans = create_model('kmeans', )
```

	Silhouette	Cainiski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.2486	2156.578	1.2769	0	0	0

```
In [ ]: print(kmeans)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
n_clusters=4, n_init=10, n_jobs=-1, precompute_distances='deprecated',
random_state=295, tol=0.0001, verbose=0)
```

```
In [ ]: plot_model(kmeans)
```

```
In [ ]: kmean_results = assign_model(kmeans)
kmean_results.head()
```

	CUST_CODE	TotalSpend	TotalQuantity	TotalVisits	TotalSKUs	MedShopHour	FirstDate	LastDate	AvgVisitsPerMonth
0	CUST0000000181	2.44	1	1	1	13.0	2007-01-06	2007-01-06	1.000000
1	CUST0000000389	959.33	923	220	189	16.0	2006-04-16	2008-07-05	7.857143
2	CUST0000000689	328.57	334	16	116	19.0	2007-07-22	2008-06-23	1.777778
3	CUST0000000998	5.95	6	3	4	11.5	2008-05-04	2008-07-07	1.500000
4	CUST0000001163	39.74	34	4	24	15.0	2008-10-22	2008-06-22	1.000000

```
In [ ]: from sklearn.model_selection import train_test_split

df_rf = kmean_results.drop(columns=['CUST_CODE', 'FirstDate', 'LastDate', 'CountShopWeekDay', 'CountShopWeekDayTotal'])
df_rf.dropna(inplace=True)
train_data = df_rf.drop(columns=['Cluster'])
train_res = df_rf['Cluster']

X_train, X_test, y_train, y_test = train_test_split(train_data, train_res, test_size=0.30, random_state=41)
```

```
In [ ]: from sklearn.preprocessing import LabelBinarizer

def multiclass_roc_auc_score(y_test, y_pred, average="macro", multi_class="ovr"):
    lb = LabelBinarizer()
    lb.fit(y_test)
    y_test = lb.transform(y_test)
    y_pred = lb.transform(y_pred)
    return roc_auc_score(y_test, y_pred, average=average, multi_class=multi_class)
```

```
In [ ]: from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import multilabel_confusion_matrix, plot_confusion_matrix, roc_auc_score, roc_curve
import matplotlib.pyplot as plt

clf = RandomForestClassifier(n_jobs=-1, random_state=41)
model = clf.fit(X_train, y_train)

lr_probs = model.predict_proba(X_test)
plot_confusion_matrix(model, X_test, y_test)
plt.show()

y_pred = model.predict(X_test)
print("AUC:", multiclass_roc_auc_score(y_test, y_pred))
```



AUC: 0.9735972471178925

```
In [ ]: fdf = pd.DataFrame(model.feature_importances_, index=X_test.columns, columns=['imp'])
fdf = fdf.sort_values(by='imp', ascending=False)

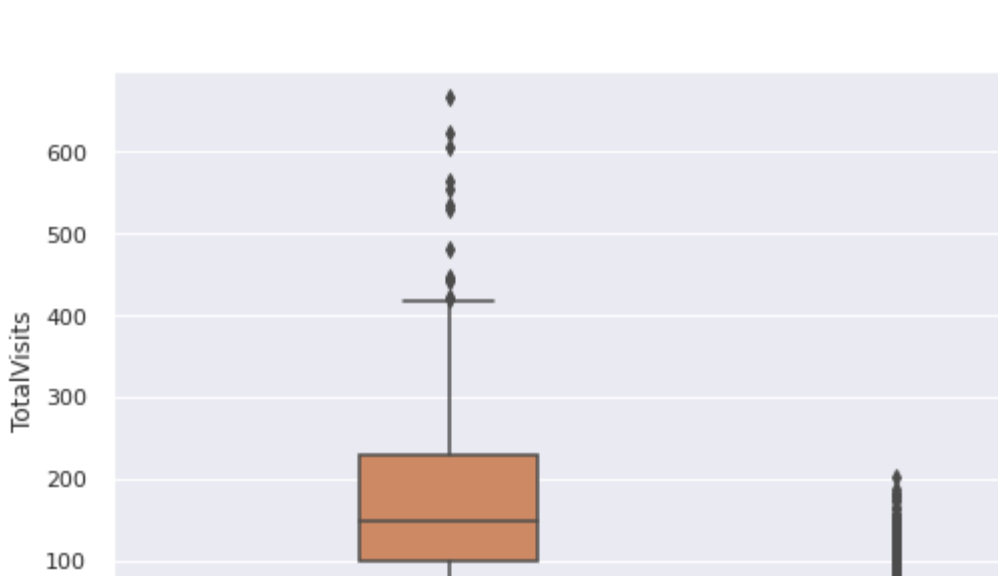
print(fdf)
```

	imp
total_days	0.170470
reecency	0.141739
TotalSpend	0.107384
TotalVisits	0.101441
TotalQuantity	0.083366
TicketSize	0.060240
ARPU	0.079351
BasketSize	0.074370
percentage_weekday	0.070135
TotalSKUs	0.045792
AvgVisitsPerMonth	0.029614
MedShopHour	0.008664
BasketDominantMode_Mixed	0.003431
BasketDominantMode_Fresh	0.002147
BasketDominantMode_Grocery	0.001157
BasketDominantMode_HomFood	0.000493
BasketDominantMode_oxx	0.000295

```
In [ ]: import seaborn as sns

sns.set(font_scale = 1)
sns.boxplot(x="Cluster", y="total_days", data=kmean_results)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe1bd4f72d0>



```
In [ ]: sns.set(font_scale = 1)
sns.boxplot(x="Cluster", y="reecency", data=kmean_results)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe1a30c4e10>



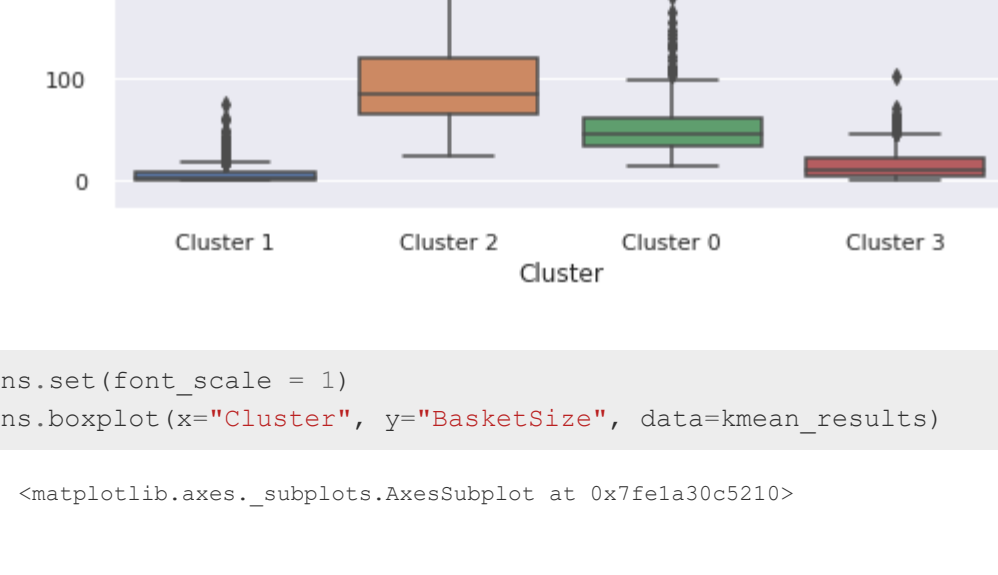
```
In [ ]: sns.set(font_scale = 1)
sns.boxplot(x="Cluster", y="TotalSpend", data=kmean_results)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe1a30c45d0>



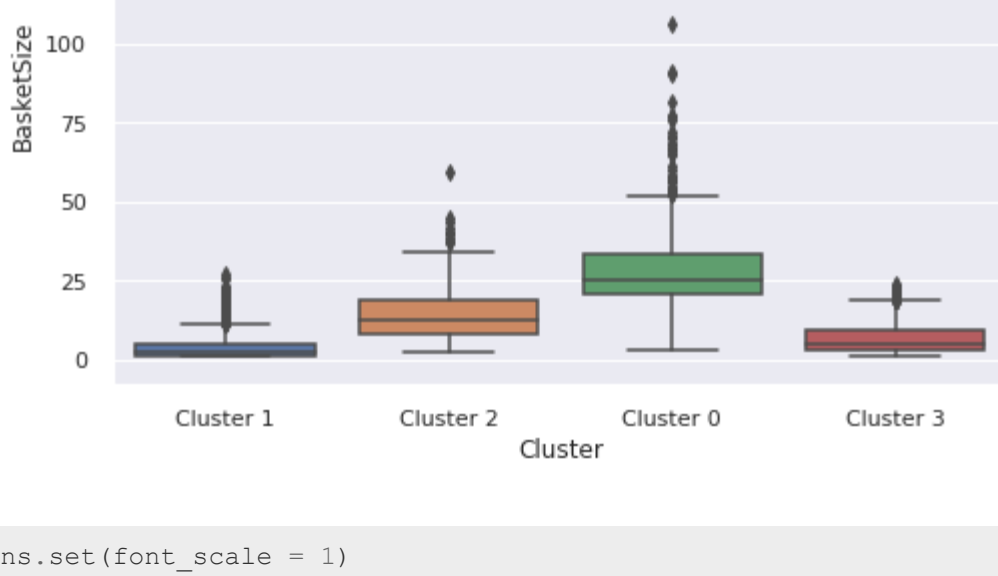
```
In [ ]: sns.set(font_scale = 1)
sns.boxplot(x="Cluster", y="TicketSize", data=kmean_results)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe1a30c62d0>



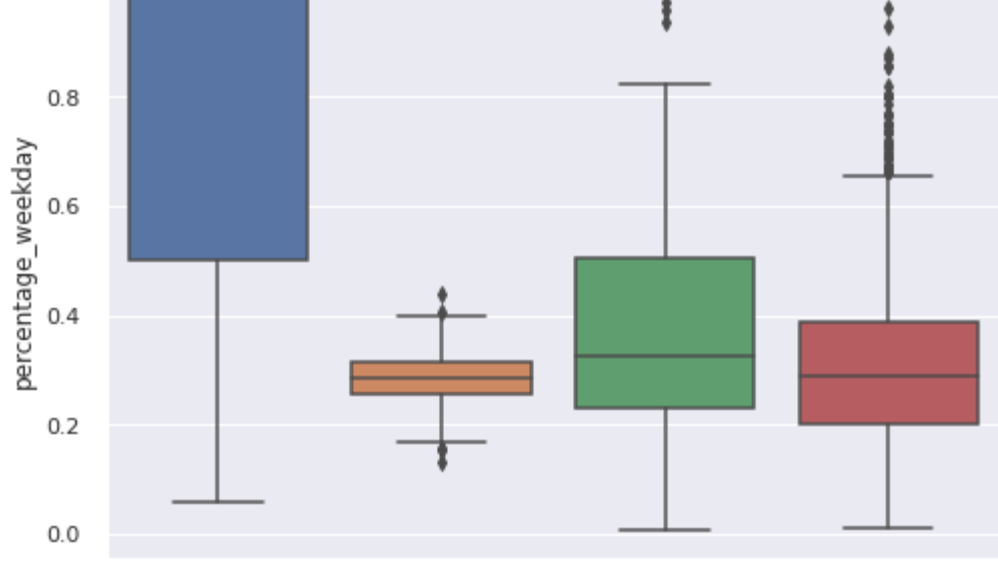
```
In [ ]: sns.set(font_scale = 1)
sns.boxplot(x="Cluster", y="ARPU", data=kmean_results)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe1a30c62d0>



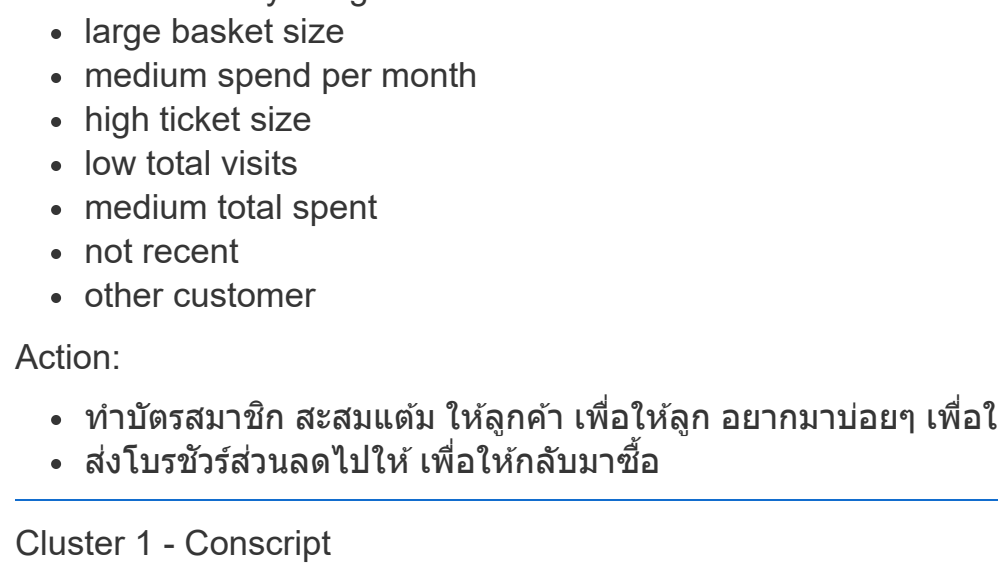
```
In [ ]: sns.set(font_scale = 1)
sns.boxplot(x="Cluster", y="BasketSize", data=kmean_results)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe1a30c62d0>



```
In [ ]: sns.set(font_scale = 1)
sns.boxplot(x="Cluster", y="percentage_weekday", data=kmean_results)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe1a30c62d0>



- Cluster 0 - MIA (Missing in action)
- low weekday = high weekend
 - large basket size
 - medium spend per month
 - high ticket size
 - low total visits
 - medium total spent
 - not recent
 - other customer

Action:

- ทำบัตรสมาชิก สะสมแต้ม ให้ลูกค้า เพื่อให้อีก ลูกค้าอยากมาบ่อยๆ เพื่อจะได้ส่วนลด หรือ สิทธิพิเศษใดๆ
- ทำโปรโมชั่นส่วนลดไปให้ เพื่อใกล้กลับมาซื้อ

Cluster 1 - Conscript

- high weekday = low weekend
- small basket size
- low spend per month
- low ticket size
- low total visits
- low total spent
- not recent
- short time customer

Action:

- ทำบัตรสมาชิก สะสมแต้ม ให้ลูกค้า เพื่อให้อีก ลูกค้าอยากมาบ่อยๆ เพื่อจะได้ส่วนลด หรือ สิทธิพิเศษใดๆ
- ทำโปรโมชั่นพิเศษวันธรรมดา เพื่อทำให้คนเข้ามาในวันธรรมดาซื้อเยอะ และบ่อยขึ้น

Cluster 2 - Commander

- low weekday = high weekend
- medium basket size
- high spend per month
- medium ticket size
- high total visits
- high total spent
- very recent
- long time customer

Action:

- ให้ Reward พิเศษ อาจจะเป็นอย่างวันธรรมดา
- ขายสินค้าที่ช่วยเพิ่มความสะดวกสบาย หรือหา

Cluster 3 - Veteran

- low weekday = high weekend
- small basket size
- low spend per month
- low ticket size
- medium total visits
- low total spent
- fairly recent
- long time customer

Action:

- ทำส่วนลดเมื่อซื้อของมากกว่าค่าเฉลี่ยเดิม ที่เขาซื้อ เพื่อ up-sale
- recommend co purchase product

```
In [ ]:
```