In [123]:

```
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
from sklearn.feature_selection import VarianceThreshold
from sklearn.preprocessing import MinMaxScaler
```

In [124]:

```
data = pd.read_csv('.\dataset_Facebook.csv')
data_new = data.values
data_new
```

Out[124]:

```
array([[139441, 'Photo', 2, ..., 79.0, 17.0, 100],
       [139441, 'Status', 2, ..., 130.0, 29.0, 164],
       [139441, 'Photo', 3, ..., 66.0, 14.0, 80],
       ...,
       [81370, 'Photo', 1, ..., 93.0, 18.0, 115],
       [81370, 'Photo', 3, ..., 91.0, 38.0, 136],
       [81370, 'Photo', 2, ..., 91.0, 28.0, 119]], dtype=object)
```

In [125]:

```
# INICIO DA ANALISE ESTATÍSTICA

#1- Quantidade de registros pagos
data[data['Paid']==1].groupby(['Paid']).agg('count').Type
#Resposta => 139 registros
```

Out[125]:

```
Paid
1.0    139
Name: Type, dtype: int64
```

In [126]:

```
#2- Ranking de tipos de postagem (Pagas)
data[data['Paid']==1].groupby(['Type']).agg('count').Paid
#Resposta =>
#
#Link 6
#Photo 119
#Status 10
#Video 1
```

Out[126]:

```
Type
Link        6
Photo     119
Status     10
Video       4
Name: Paid, dtype: int64
```

In [127]:

```
#3- Quais os três meses que teve mais publicações
data.groupby(['Post Month']).agg('count').sort_values(by=['Type'], ascending=False).Type.he
#Resposta =>
#Outubro = 60
#Julho = 52
#Abril = 50
```

Out[127]:

```
Post Month
10    60
7     52
4     50
Name: Type, dtype: int64
```

In [128]:

```
#4- Os cinco melhores horários para publicações pagas
data[data['Paid']==1].groupby(['Post Hour']).agg('count').sort_values(by=['Type'], ascendir
# Respostas =>
#3h = 32 publicações
#10h = 18 publicações
#13h = 15 publicações
#2h = 14 publicações
#11h = 11 publicações
```

Out[128]:

```
Post Hour
3     32
10    18
13    15
2     14
11    11
Name: Type, dtype: int64
```

In [129]:

```
#5- O horário do dia com mais publicação de fotos
data[data['Type']=='Photo'].groupby(['Post Hour']).agg('count').sort_values(by=['Type'], as
#Resposta =>
#3h => 89 publicações de fotos
```

Out[129]:

```
Post Hour
3    89
Name: Type, dtype: int64
```

In [130]:

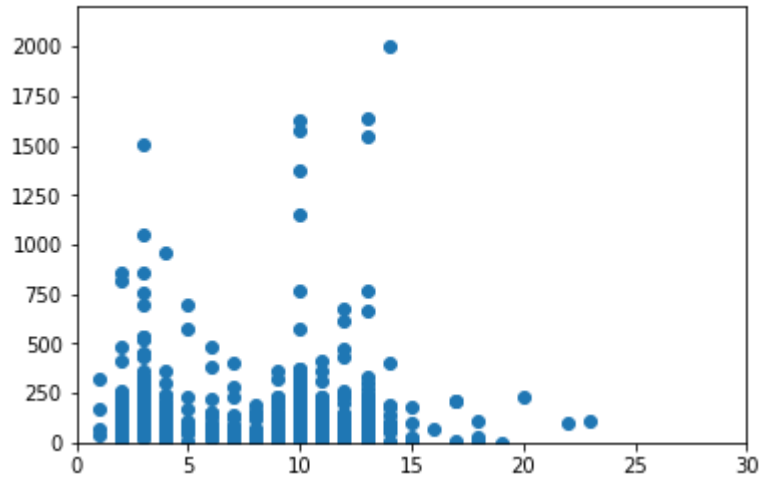
```
# FIM DA ANALISE ESTATÍSTICA
```

In [131]:

```
# INICIO DA SELEÇÃO DE ATRIBUTOS / REDUÇÃO DE DIMENSÃO
data_filtered = []
idPhoto = -1
for item in data_new:
    if (not np.isnan(item[0])) and (not np.isnan(item[2])) and (not np.isnan(item[3])) and
        if (item[1] == 'Photo'):
            idPhoto = 0
        elif (item[1] == 'Status'):
            idPhoto = 1
        elif (item[1] == 'Video'):
            idPhoto = 2
        elif (item[1] == 'Link'):
            idPhoto = 3
        item[1] = idPhoto
        data_filtered.append(item[0:19])

xx = []
yy = []
for value in data_filtered:
    xx.append(value[5])
    yy.append(value[16])

plt.xlim([0,30])
plt.ylim([0,2200])
plt.scatter(xx,yy)
plt.show()
```



In [132]:

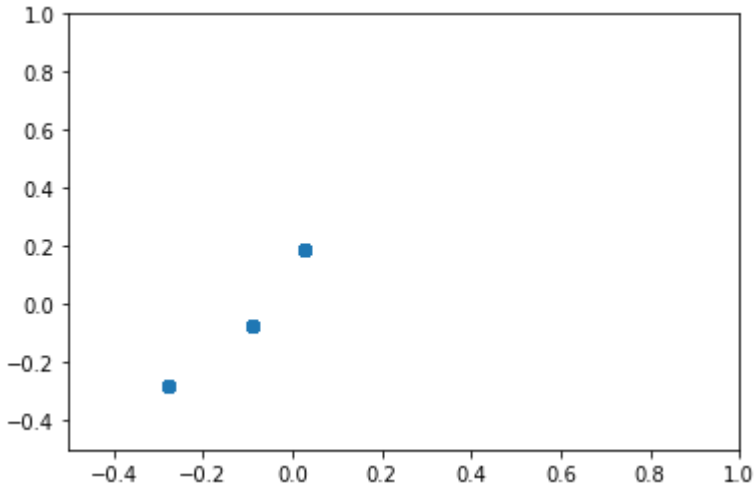
```
min_max_scaler = MinMaxScaler()
seletor_atributos = VarianceThreshold(threshold=.08)
data_norm = min_max_scaler.fit_transform(data_filtered)
data_result = seletor_atributos.fit_transform(data_norm)
data_result
```

Out[132]:

```
array([[0.5       , 1.        , 0.5       , 0.        ],
       [0.5       , 1.        , 0.33333333, 0.        ],
       [1.        , 1.        , 0.33333333, 0.        ],
       ...,
       [0.5       , 0.        , 0.66666667, 0.        ],
       [0.        , 0.        , 0.66666667, 0.        ],
       [1.        , 0.        , 0.5       , 0.        ]])
```

In [133]:

```
pca = PCA(n_components=3)
m_out = pca.fit_transform(data_result)
zz = []
kk = []
for value in m_out:
    zz.append(m_out[0])
    kk.append(m_out[1])
plt.xlim([-0.5,1])
plt.ylim([-0.5,1])
plt.scatter(zz,kk)
plt.show()
```



In [134]:

```
# Inicialmente foi verificado que havia necessidade de transformar a coluna Type que era um
# Foi realizado mediante tratativa da variavel IdPhoto.
# Em seguida foi verificado que existiam valores NaN. Para isso foi realizado
# o tratamento para remover a mesma do array data_filtered
# Após isso um gráfico para auxiliar a visualização.
# Feito isso, aplicado uma variancia de 0.08 e o dataset reduziu de 20 para 4 colunas.
# Por fim, aplicado o PCA de 3 colunas e exibido um gráfico.
# É possível verificar que há apenas 3 pontos crescentes onde há concentração de valores.
```

In [ ]:

```
```