Libs

```
import arrow
from icecream import ic
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
from pandas.tseries.holiday import AbstractHolidayCalendar, Holiday,
MO, USFederalHolidayCalendar
from prophet import Prophet
```

```
C:\Users\DV\AppData\Local\Programs\Python\Python310\lib\site-packages\
tqdm\auto.py:22: TqdmWarning: IProgress not found. Please update
jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

Load Data

```
data = pd.read_csv("data_wf.csv")
data.head()
```

```
       Date  No_of_calls_from_commercial_card
No_of_calls_from_retail_card  \
0  1/1/2017                              264.0
403.0
1  1/2/2017                              528.0
792.0
2  1/3/2017                             2505.0
3774.0
3  1/4/2017                             2145.0
3236.0
4  1/5/2017                             2164.0
3265.0

   No_of_calls_from_student_card
0                          860.0
1                         1681.0
2                         7639.0
3                         6513.0
4                         6604.0
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1765 entries, 0 to 1764
Data columns (total 4 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
```

```
 0   Date                                1673 non-null   object
 1   No_of_calls_from_commercial_card  1581 non-null   float64
 2   No_of_calls_from_retail_card      1581 non-null   float64
 3   No_of_calls_from_student_card     1581 non-null   float64
dtypes: float64(3), object(1)
memory usage: 55.3+ KB
```

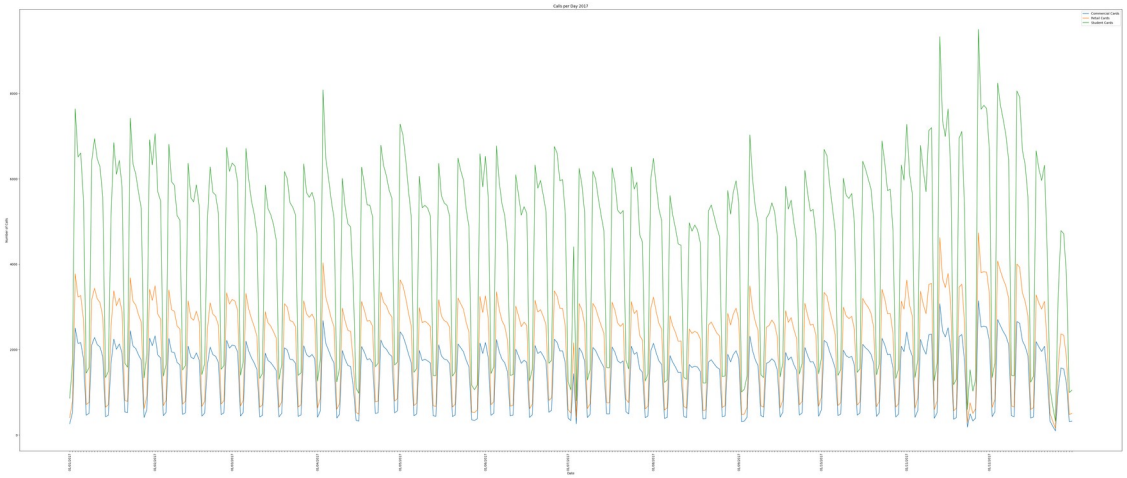
EDA

```
# Copy Data
df = data.copy()

df = df.dropna(how='any')

df.rename(columns={"Date":"date",
"No_of_calls_from_commercial_card":"cc",
"No_of_calls_from_retail_card":"rc",
"No_of_calls_from_student_card":"sc"}, inplace=True)

df['date'] = df['date'].apply(lambda row: arrow.get(row,
"M/D/YYYY").format("DD/MM/YYYY"))

df['day'] = df['date'].apply(lambda row: arrow.get(row,
"DD/MM/YYYY").day)
df['month'] = df['date'].apply(lambda row: arrow.get(row,
"DD/MM/YYYY").month)
df['year'] = df['date'].apply(lambda row: arrow.get(row,
"DD/MM/YYYY").year)

df.head()
```

	date	cc	rc	sc	day	month	year
0	01/01/2017	264.0	403.0	860.0	1	1	2017
1	02/01/2017	528.0	792.0	1681.0	2	1	2017
2	03/01/2017	2505.0	3774.0	7639.0	3	1	2017
3	04/01/2017	2145.0	3236.0	6513.0	4	1	2017
4	05/01/2017	2164.0	3265.0	6604.0	5	1	2017

```
# 2017 Dataframe
df_17 = df[df['year'] == 2017]

# Plot calls per day for Commercial, Retail, and Student Cards
x = range(len(df_17['date']))
cc_values = df_17.cc
rc_values = df_17.rc
sc_values = df_17.sc
fig = plt.figure()
fig.set_figheight(20)
fig.set_figwidth(50)
ax = fig.add_axes([1,1,1,1])
ax.plot(x, cc_values, label='Commercial Cards')
```

```
ax.plot(x, rc_values, label='Retail Cards')
ax.plot(x, sc_values, label='Student Cards')
ax.set_xticks(x, [ dt if arrow.get(dt, "DD/MM/YYYY").day==1 else ""
for dt in df_17['date'] ], rotation=90)
ax.set_xlabel("Date")
ax.set_ylabel("Number of Calls")
ax.set_title("Calls per Day 2017")
ax.legend()
plt.show()
```

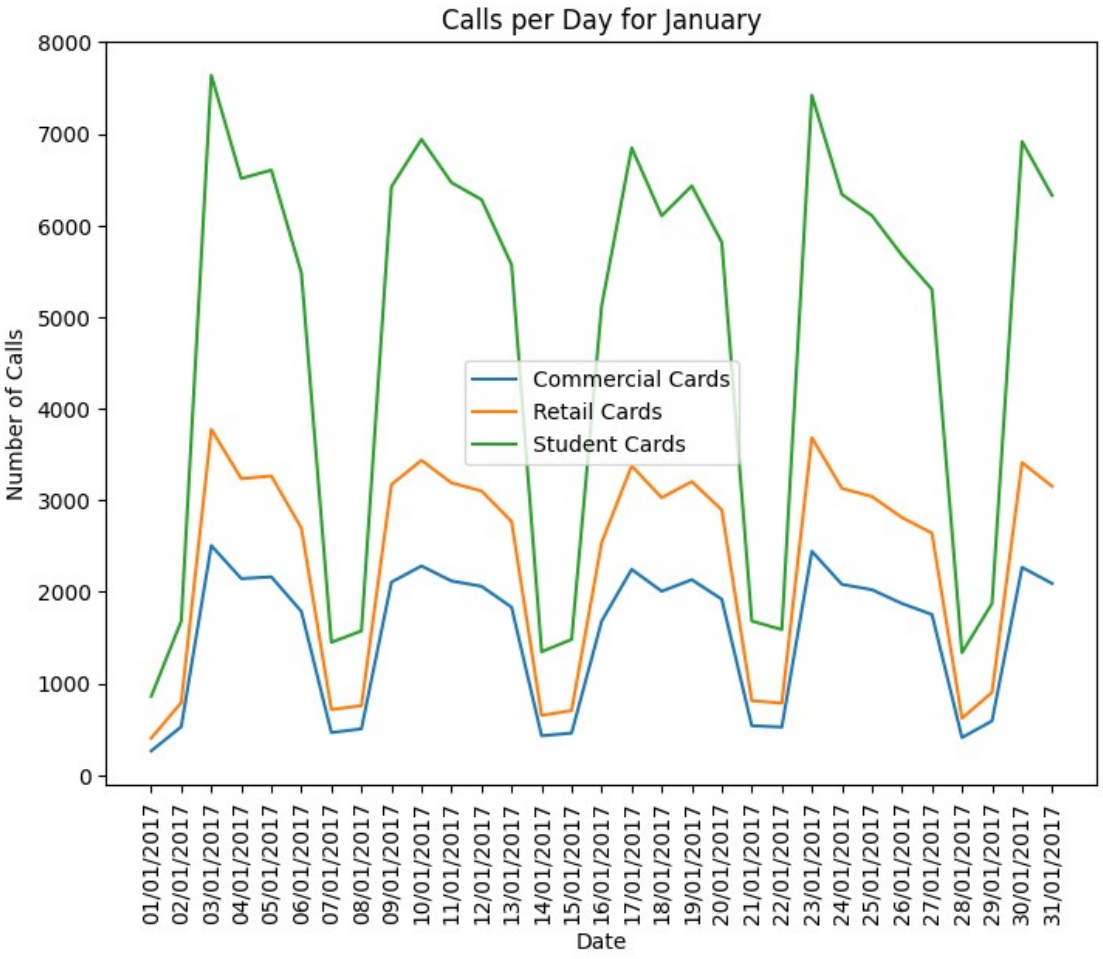


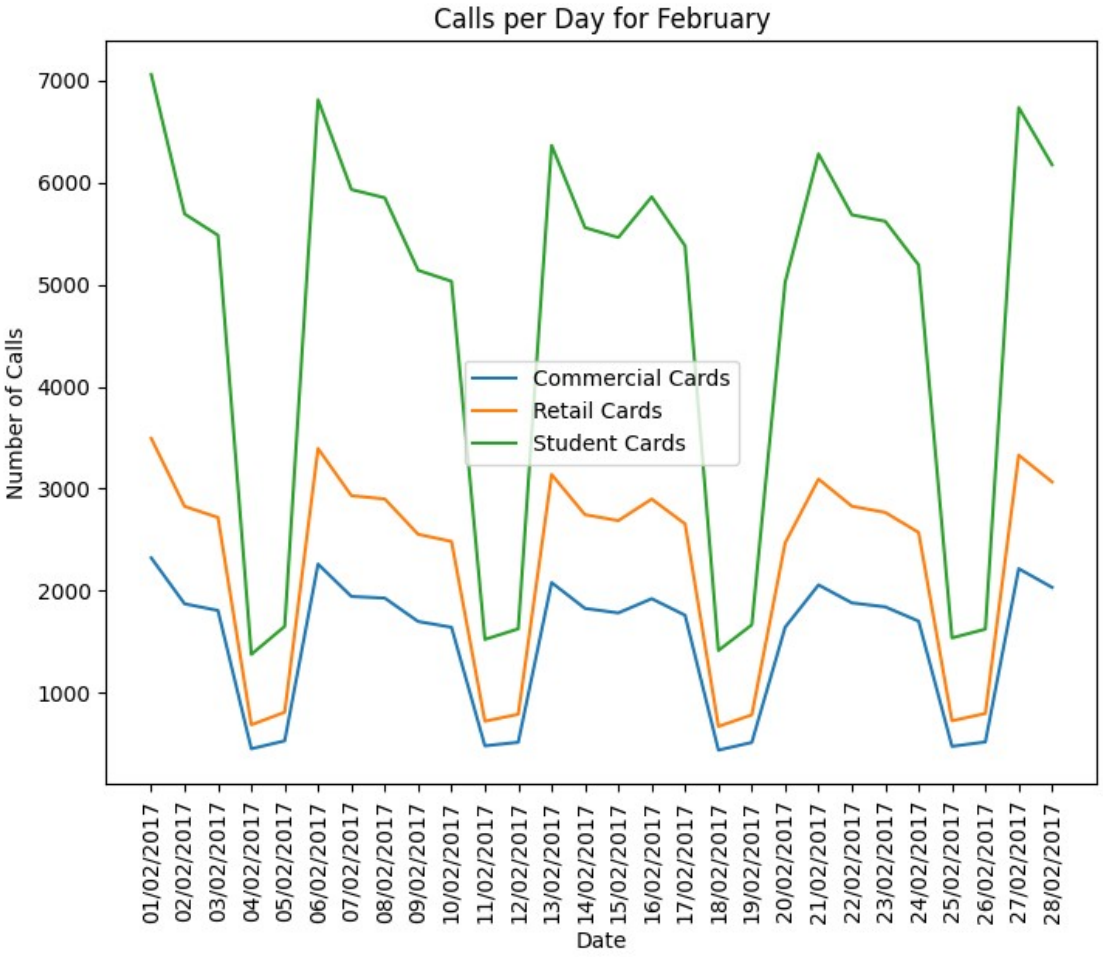
```
# Trend Observed
# cc is lowest for every observation
# rc is middle
# sc is highest
obs = df_17.shape[0]
print(((df_17.cc < df_17.rc) & (df_17.rc < df_17.sc)).sum() == obs)
```

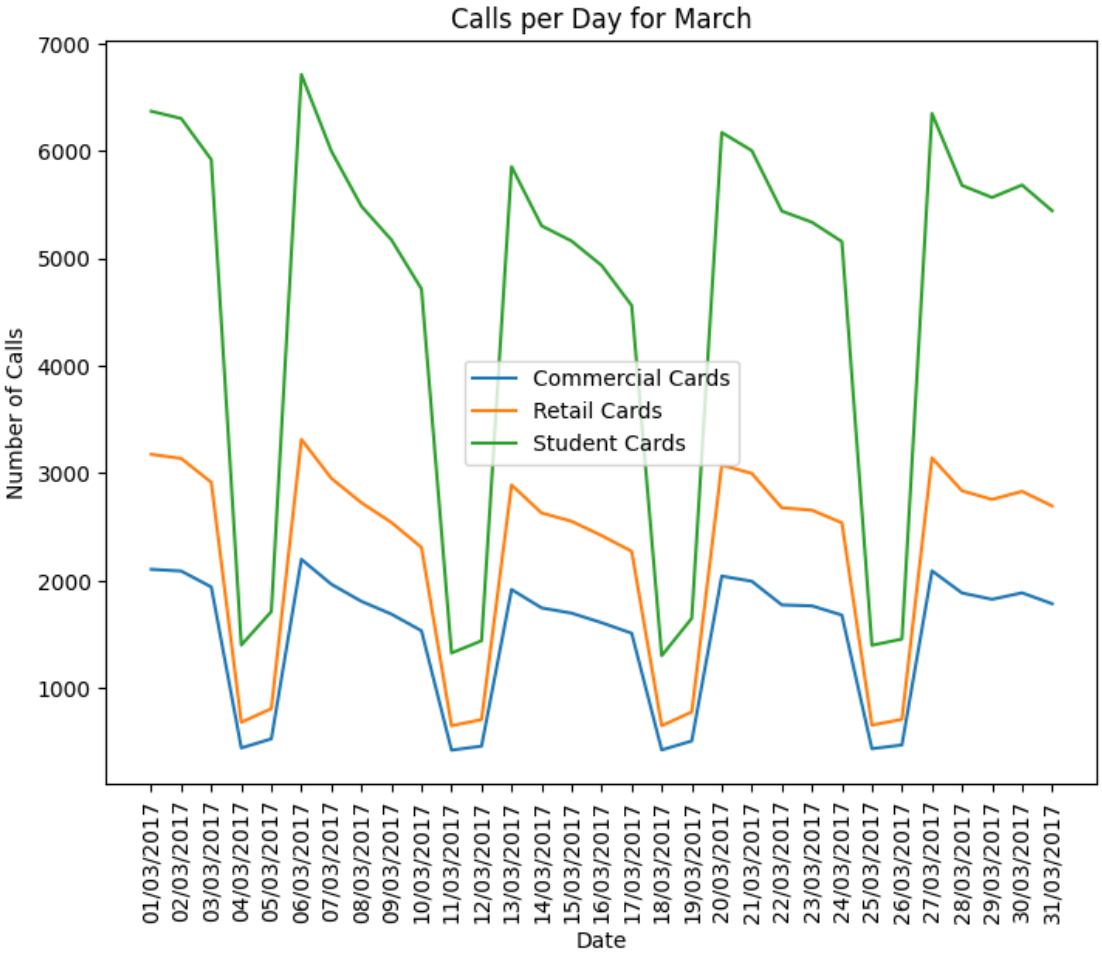
```
True
```

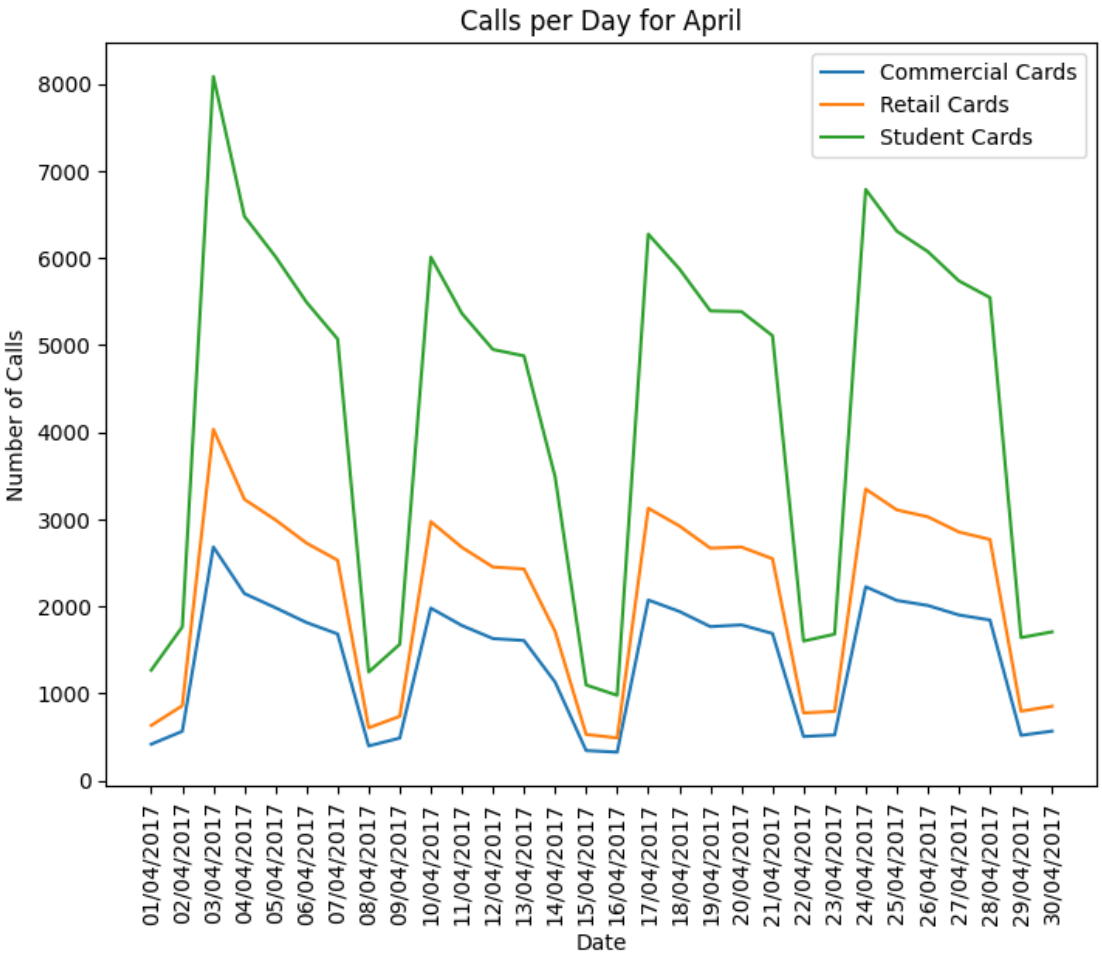
```
# 2017 data for each month
# Low number of calls on weekends
# Low number of calls on Big Holidays
months = df_17.month.unique()
for m in months:
    x = range((df_17.month == m).sum())
    cc_values = df_17[df_17.month == m].cc
    rc_values = df_17[df_17.month == m].rc
    sc_values = df_17[df_17.month == m].sc
    fig = plt.figure()
    ax = fig.add_axes([1,1,1,1])
    ax.plot(x, cc_values, label='Commercial Cards')
    ax.plot(x, rc_values, label='Retail Cards')
    ax.plot(x, sc_values, label='Student Cards')
    ax.set_xticks(x, df_17[df_17.month == m].date, rotation=90)
    ax.set_xlabel("Date")
    ax.set_ylabel("Number of Calls")
```

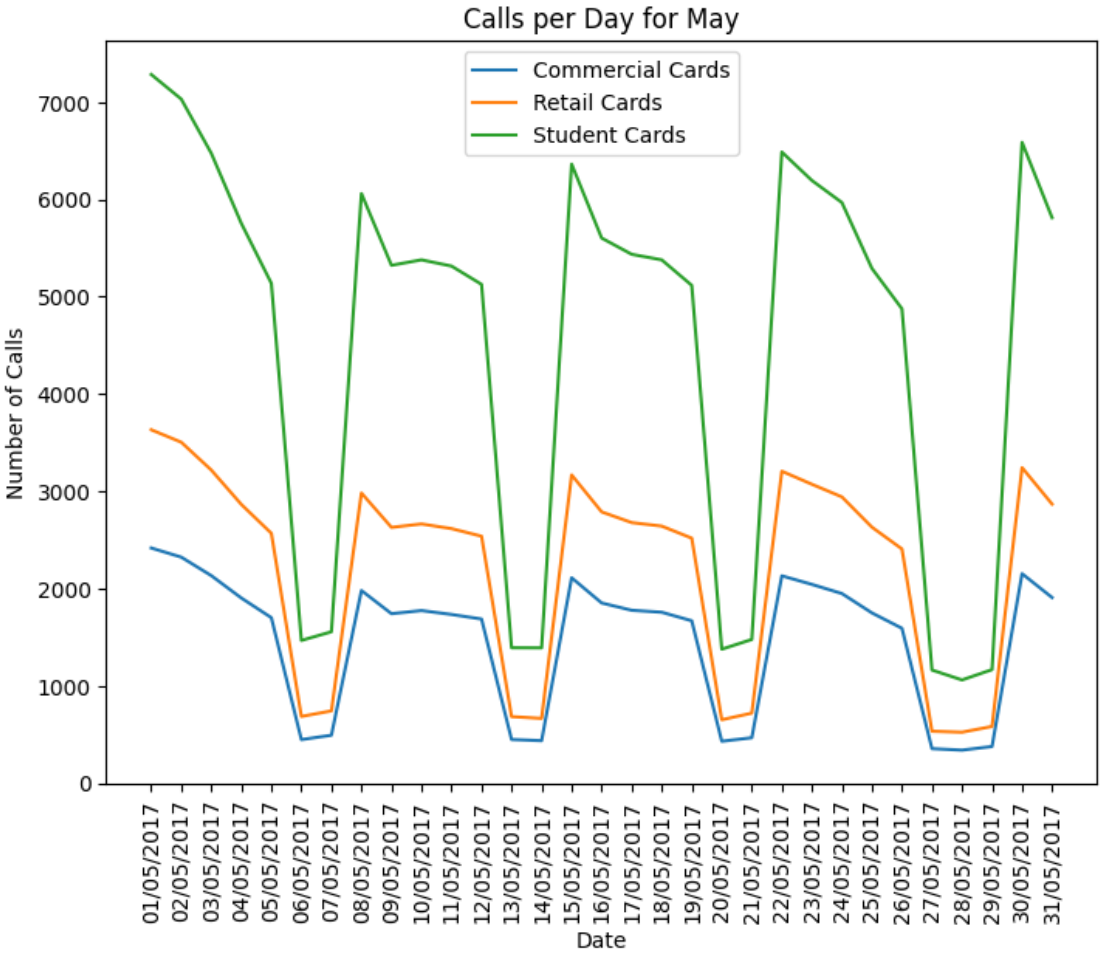
```
    ax.set_title(f"Calls per Day for {arrow.get(str(m),
'M').format('MMMM')}")
    ax.legend()
    plt.show()
```

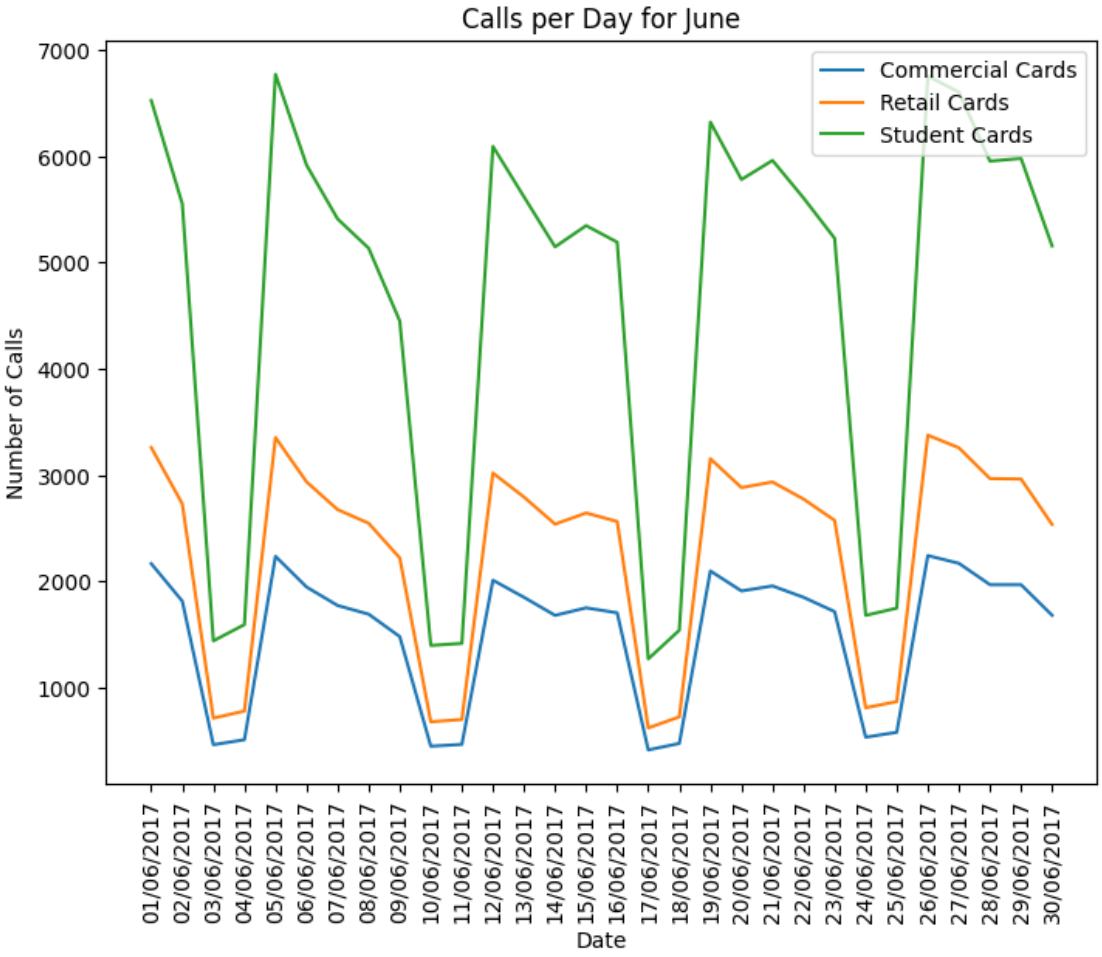


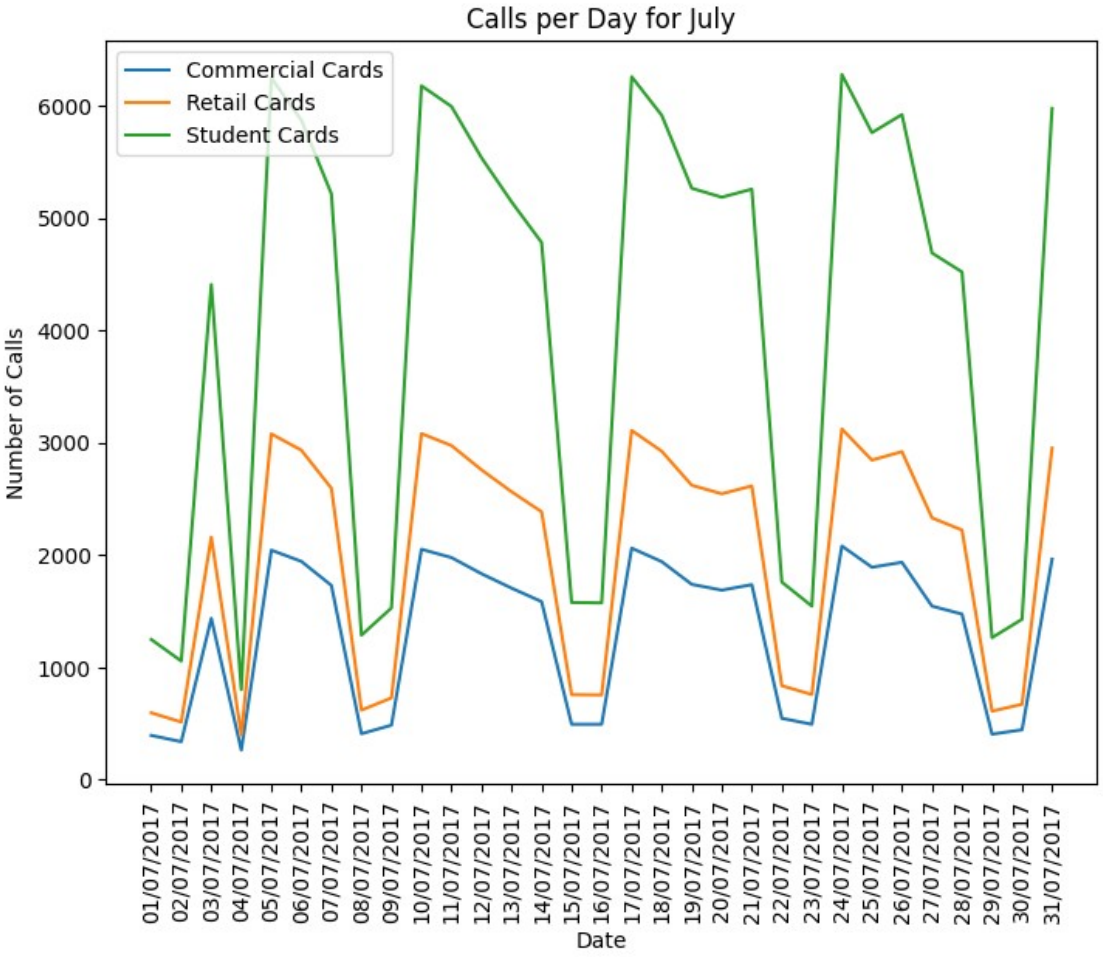

Calls per Day for February
Number of Calls
7000
6000
5000
4000
3000
2000
1000
Commercial Cards
Retail Cards
Student Cards
01/02/2017
02/02/2017
03/02/2017
04/02/2017
05/02/2017
06/02/2017
07/02/2017
08/02/2017
09/02/2017
10/02/2017
11/02/2017
12/02/2017
13/02/2017
14/02/2017
15/02/2017
16/02/2017
17/02/2017
18/02/2017
19/02/2017
20/02/2017
21/02/2017
22/02/2017
23/02/2017
24/02/2017
25/02/2017
26/02/2017
27/02/2017
28/02/2017
Date

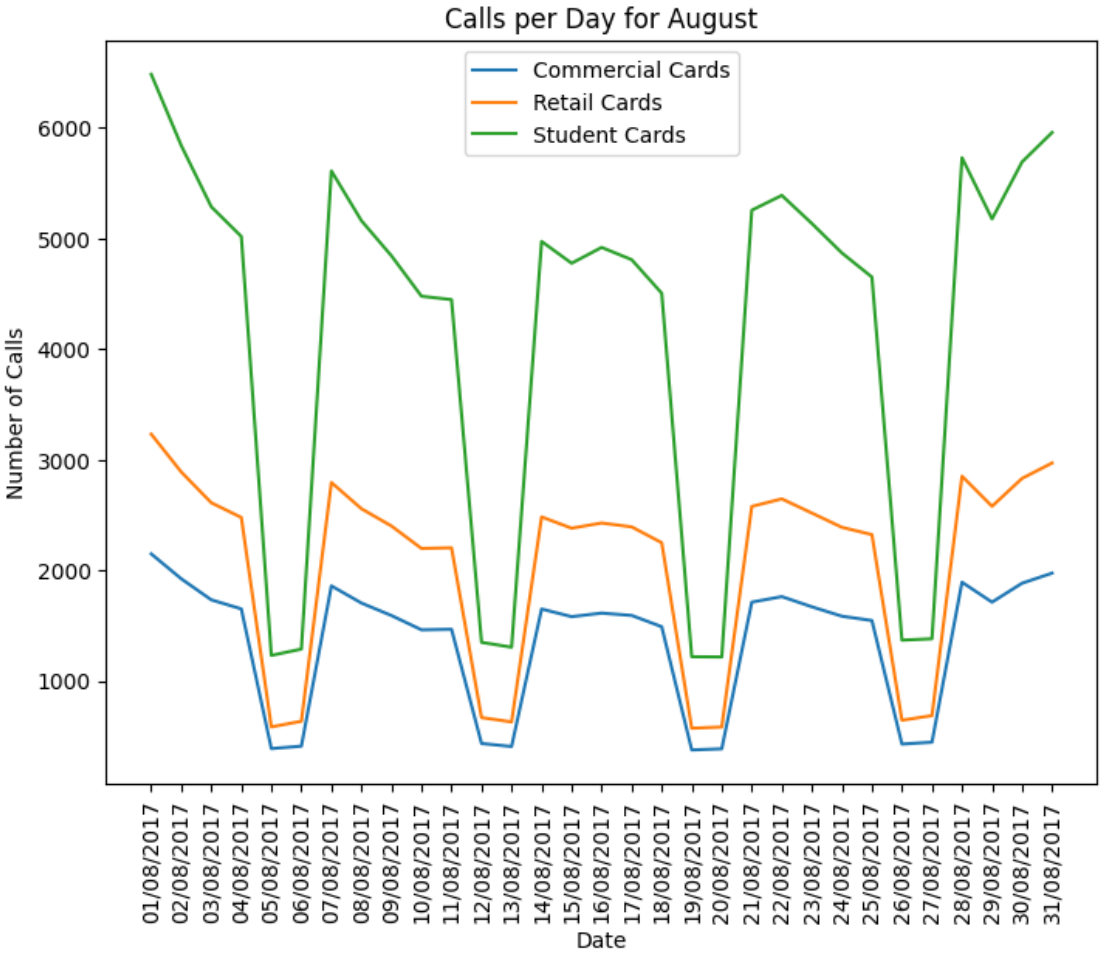

Calls per Day for March
Number of Calls
7000
6000
5000
4000
3000
2000
1000
Commercial Cards
Retail Cards
Student Cards
01/03/2017
02/03/2017
03/03/2017
04/03/2017
05/03/2017
06/03/2017
07/03/2017
08/03/2017
09/03/2017
10/03/2017
11/03/2017
12/03/2017
13/03/2017
14/03/2017
15/03/2017
16/03/2017
17/03/2017
18/03/2017
19/03/2017
20/03/2017
21/03/2017
22/03/2017
23/03/2017
24/03/2017
25/03/2017
26/03/2017
27/03/2017
28/03/2017
29/03/2017
30/03/2017
31/03/2017
Date

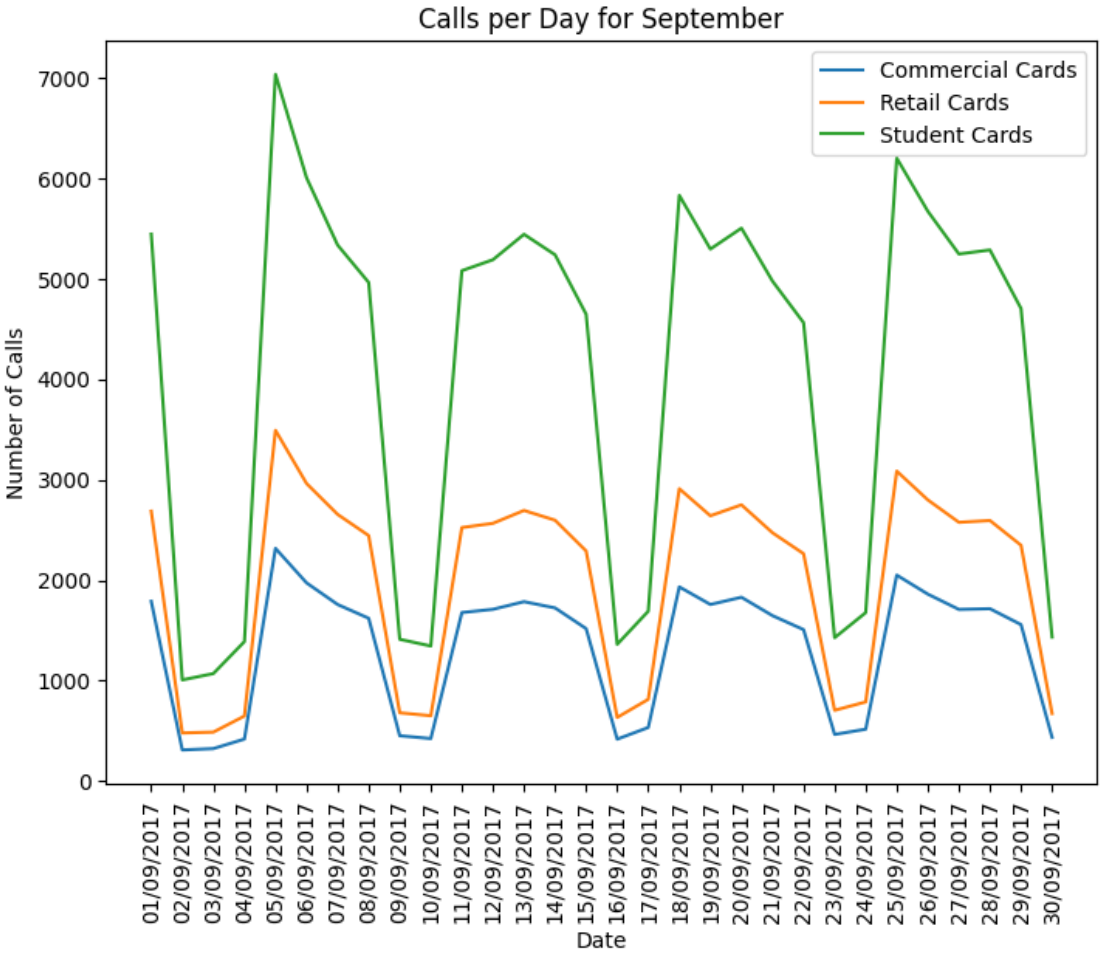

Calls per Day for April
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
8000
01/04/2017
02/04/2017
03/04/2017
04/04/2017
05/04/2017
06/04/2017
07/04/2017
08/04/2017
09/04/2017
10/04/2017
11/04/2017
12/04/2017
13/04/2017
14/04/2017
15/04/2017
16/04/2017
17/04/2017
18/04/2017
19/04/2017
20/04/2017
21/04/2017
22/04/2017
23/04/2017
24/04/2017
25/04/2017
26/04/2017
27/04/2017
28/04/2017
29/04/2017
30/04/2017
Date

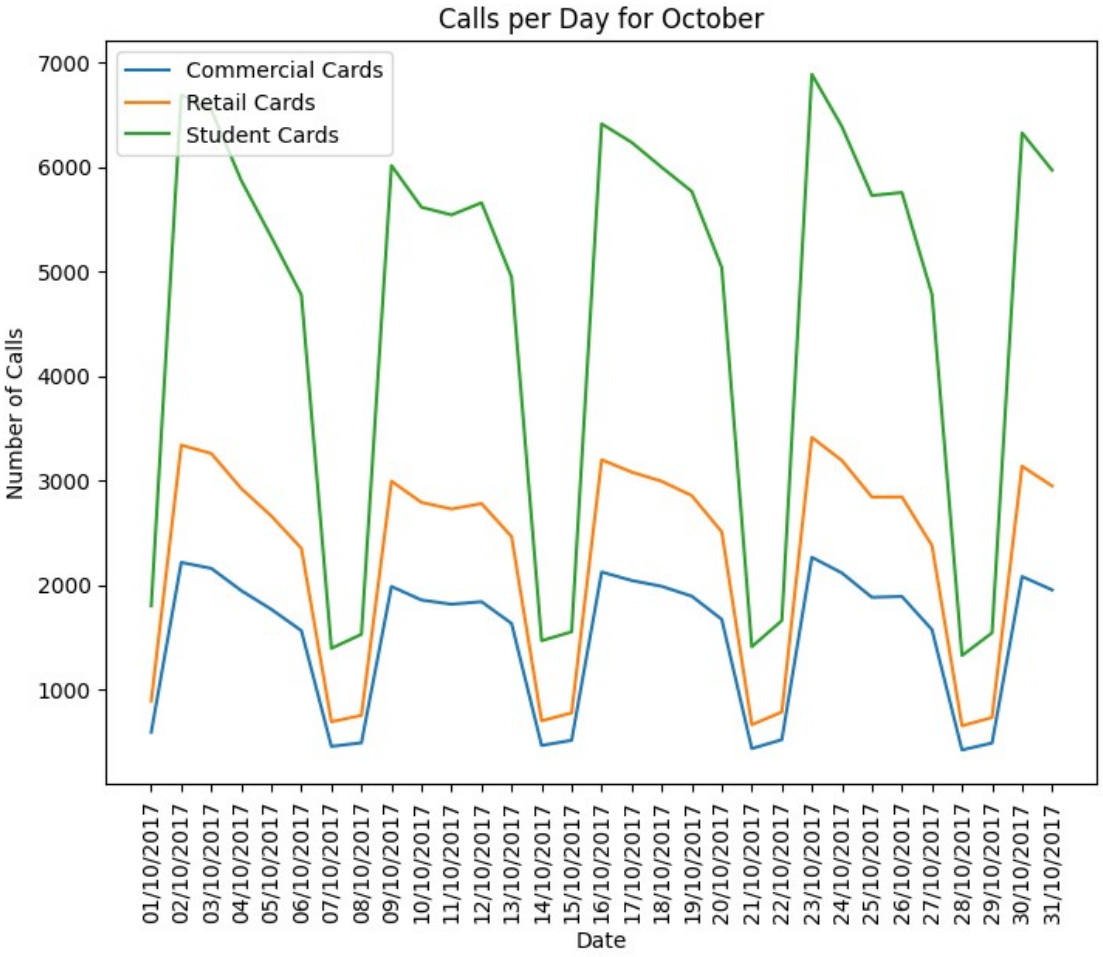

Calls per Day for May
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/05/2017
02/05/2017
03/05/2017
04/05/2017
05/05/2017
06/05/2017
07/05/2017
08/05/2017
09/05/2017
10/05/2017
11/05/2017
12/05/2017
13/05/2017
14/05/2017
15/05/2017
16/05/2017
17/05/2017
18/05/2017
19/05/2017
20/05/2017
21/05/2017
22/05/2017
23/05/2017
24/05/2017
25/05/2017
26/05/2017
27/05/2017
28/05/2017
29/05/2017
30/05/2017
31/05/2017
Date

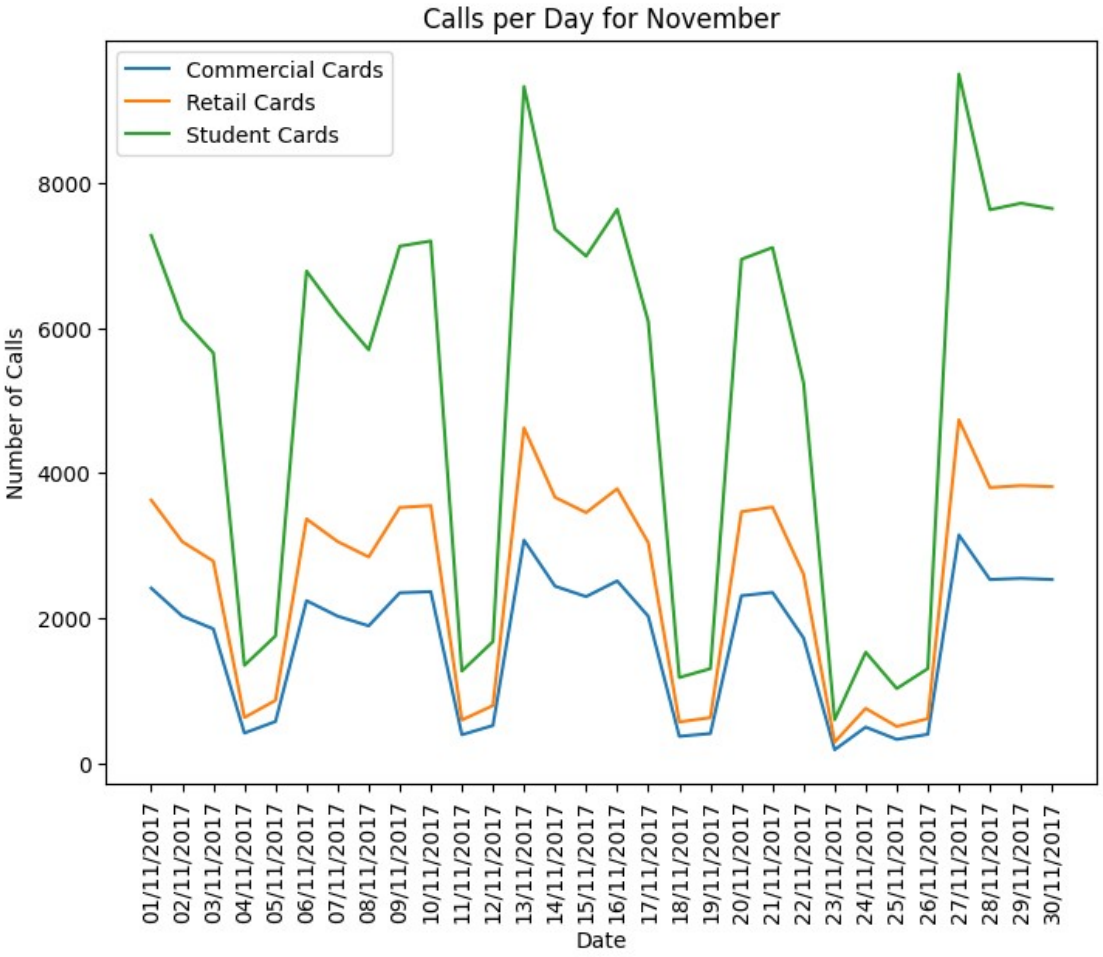

Calls per Day for June
Commercial Cards
Retail Cards
Student Cards
Number of Calls
7000
6000
5000
4000
3000
2000
1000
01/06/2017
02/06/2017
03/06/2017
04/06/2017
05/06/2017
06/06/2017
07/06/2017
08/06/2017
09/06/2017
10/06/2017
11/06/2017
12/06/2017
13/06/2017
14/06/2017
15/06/2017
16/06/2017
17/06/2017
18/06/2017
19/06/2017
20/06/2017
21/06/2017
22/06/2017
23/06/2017
24/06/2017
25/06/2017
26/06/2017
27/06/2017
28/06/2017
29/06/2017
30/06/2017
Date

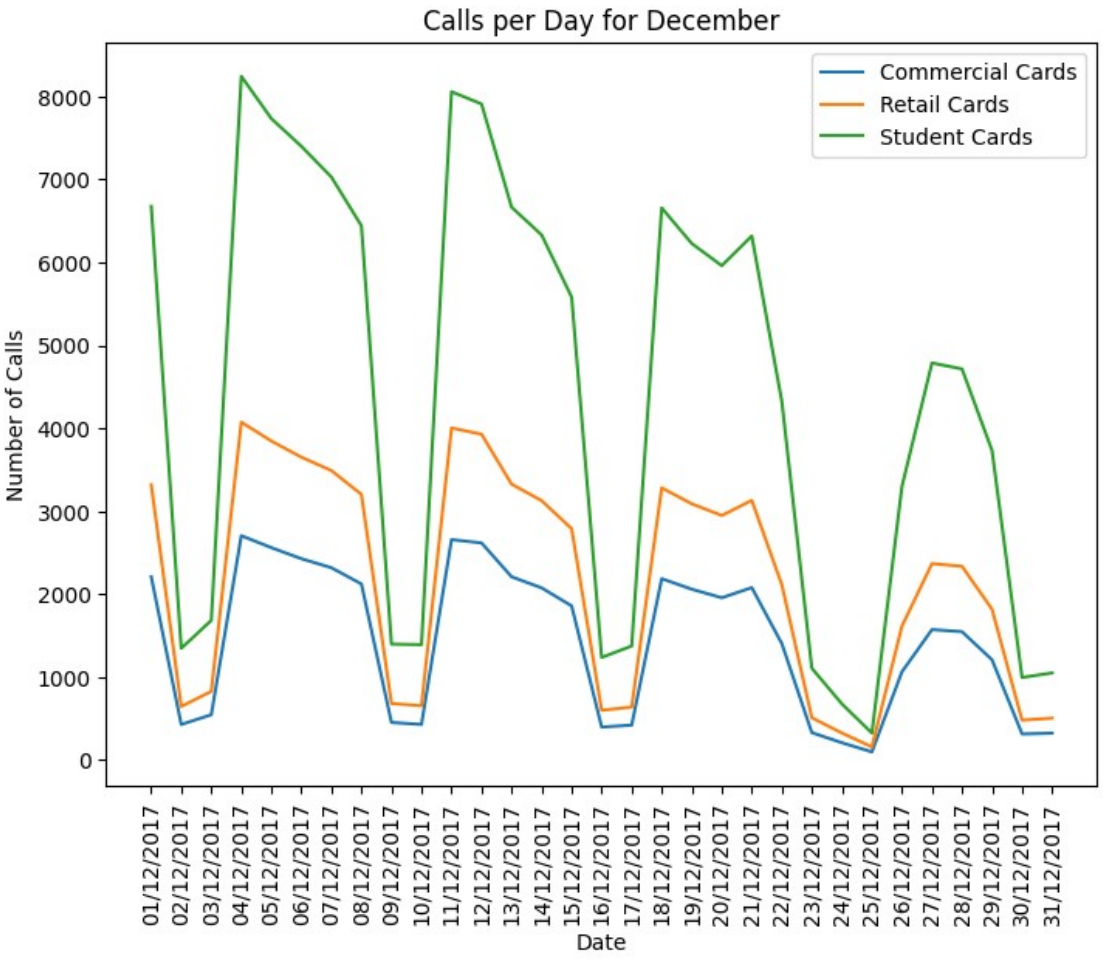

Calls per Day for July
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
01/07/2017
02/07/2017
03/07/2017
04/07/2017
05/07/2017
06/07/2017
07/07/2017
08/07/2017
09/07/2017
10/07/2017
11/07/2017
12/07/2017
13/07/2017
14/07/2017
15/07/2017
16/07/2017
17/07/2017
18/07/2017
19/07/2017
20/07/2017
21/07/2017
22/07/2017
23/07/2017
24/07/2017
25/07/2017
26/07/2017
27/07/2017
28/07/2017
29/07/2017
30/07/2017
31/07/2017
Date


Calls per Day for August
Commercial Cards
Retail Cards
Student Cards
Number of Calls
6000
5000
4000
3000
2000
1000
01/08/2017
02/08/2017
03/08/2017
04/08/2017
05/08/2017
06/08/2017
07/08/2017
08/08/2017
09/08/2017
10/08/2017
11/08/2017
12/08/2017
13/08/2017
14/08/2017
15/08/2017
16/08/2017
17/08/2017
18/08/2017
19/08/2017
20/08/2017
21/08/2017
22/08/2017
23/08/2017
24/08/2017
25/08/2017
26/08/2017
27/08/2017
28/08/2017
29/08/2017
30/08/2017
31/08/2017
Date


Calls per Day for September
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/09/2017
02/09/2017
03/09/2017
04/09/2017
05/09/2017
06/09/2017
07/09/2017
08/09/2017
09/09/2017
10/09/2017
11/09/2017
12/09/2017
13/09/2017
14/09/2017
15/09/2017
16/09/2017
17/09/2017
18/09/2017
19/09/2017
20/09/2017
21/09/2017
22/09/2017
23/09/2017
24/09/2017
25/09/2017
26/09/2017
27/09/2017
28/09/2017
29/09/2017
30/09/2017
Date


Calls per Day for October
Commercial Cards
Retail Cards
Student Cards
Number of Calls
7000
6000
5000
4000
3000
2000
1000
01/10/2017
02/10/2017
03/10/2017
04/10/2017
05/10/2017
06/10/2017
07/10/2017
08/10/2017
09/10/2017
10/10/2017
11/10/2017
12/10/2017
13/10/2017
14/10/2017
15/10/2017
16/10/2017
17/10/2017
18/10/2017
19/10/2017
20/10/2017
21/10/2017
22/10/2017
23/10/2017
24/10/2017
25/10/2017
26/10/2017
27/10/2017
28/10/2017
29/10/2017
30/10/2017
31/10/2017
Date


Calls per Day for November
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
2000
4000
6000
8000
01/11/2017
02/11/2017
03/11/2017
04/11/2017
05/11/2017
06/11/2017
07/11/2017
08/11/2017
09/11/2017
10/11/2017
11/11/2017
12/11/2017
13/11/2017
14/11/2017
15/11/2017
16/11/2017
17/11/2017
18/11/2017
19/11/2017
20/11/2017
21/11/2017
22/11/2017
23/11/2017
24/11/2017
25/11/2017
26/11/2017
27/11/2017
28/11/2017
29/11/2017
30/11/2017
Date




```
# 2018-2021 Dataframe
df_not17 = df[df['year'] != 2017]

df_not17.reset_index(drop=True, inplace=True)

df_not17.head()
```

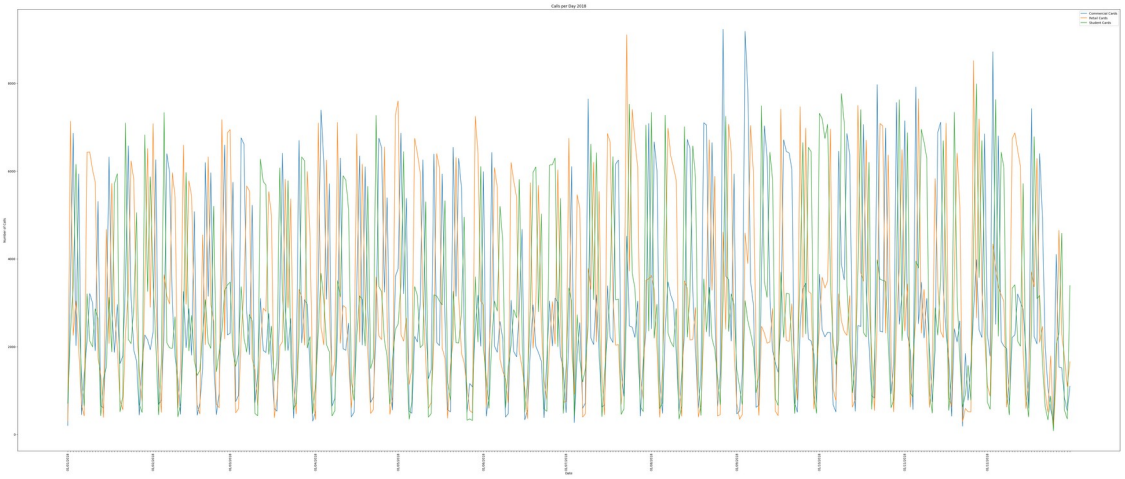
	date	cc	rc	sc	day	month	year
0	01/01/2018	200.0	305.0	709.0	1	1	2018
1	02/01/2018	3529.0	7136.0	2343.0	2	1	2018
2	03/01/2018	6862.0	2257.0	3404.0	3	1	2018
3	04/01/2018	2024.0	3036.0	6153.0	4	1	2018
4	05/01/2018	5934.0	1970.0	2961.0	5	1	2018

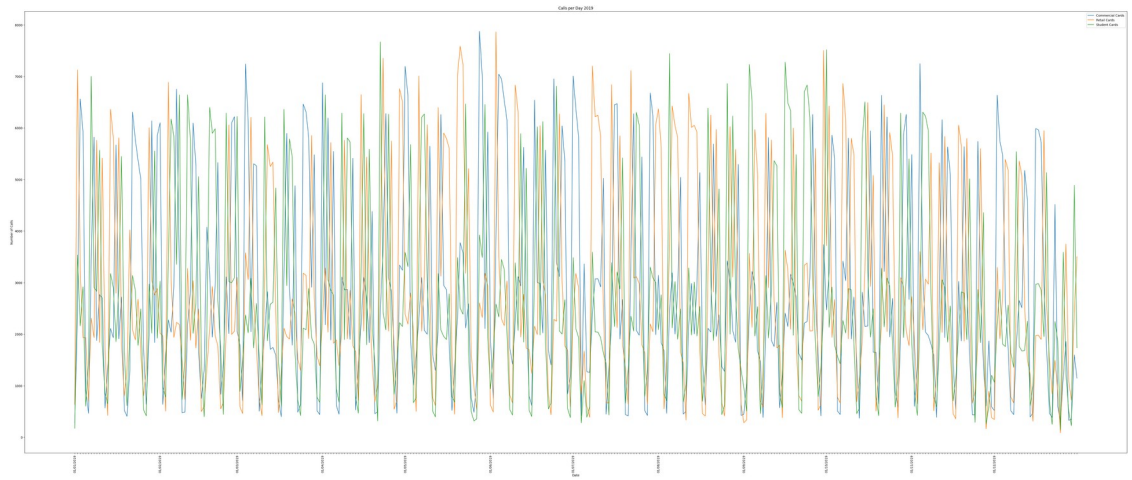
```
# Sorting data according to trend -> (cc < rc < sc) for every day
observed
df_not17_sorted = pd.DataFrame(columns=['date', 'cc', 'rc', 'sc',
'day', 'month', 'year'])
for index, row in df_not17.iterrows():
    values = sorted([row.cc, row.rc, row.sc])
    row_data = [row.date]
    row_data.extend(values)
```

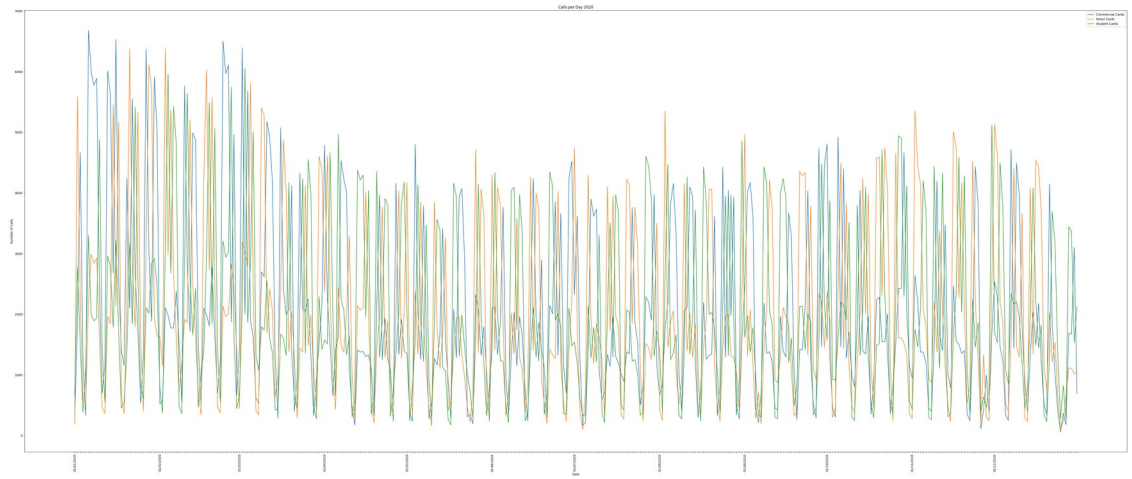
```
    row_data.extend([row.day, row.month, row.year])
    df_not17_sorted.loc[index] = row_data

df_not17_sorted.head()
```

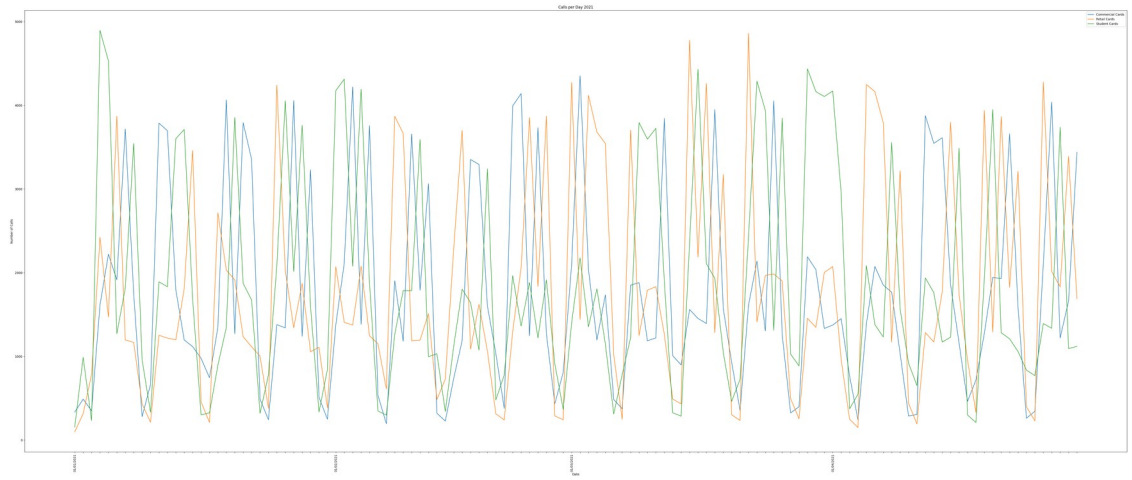
	date	cc	rc	sc	day	month	year
0	01/01/2018	200.0	305.0	709.0	1	1	2018
1	02/01/2018	2343.0	3529.0	7136.0	2	1	2018
2	03/01/2018	2257.0	3404.0	6862.0	3	1	2018
3	04/01/2018	2024.0	3036.0	6153.0	4	1	2018
4	05/01/2018	1970.0	2961.0	5934.0	5	1	2018

```
# Plot unsorted data
years = df_not17.year.unique()
for y in years:
    x = range((df_not17['year'] == y).sum())
    cc_values = df_not17[df_not17.year == y].cc
    rc_values = df_not17[df_not17.year == y].rc
    sc_values = df_not17[df_not17.year == y].sc
    fig = plt.figure()
    fig.set_figheight(20)
    fig.set_figwidth(50)
    ax = fig.add_axes([1,1,1,1])
    ax.plot(x, cc_values, label='Commercial Cards')
    ax.plot(x, rc_values, label='Retail Cards')
    ax.plot(x, sc_values, label='Student Cards')
    ax.set_xticks(x, [ dt if arrow.get(dt, "DD/MM/YYYY").day==1 else
"" for dt in df_not17[df_not17.year == y]['date'] ], rotation=90)
    ax.set_xlabel("Date")
    ax.set_ylabel("Number of Calls")
    ax.set_title(f"Calls per Day {y}")
    ax.legend()
    plt.show()
```



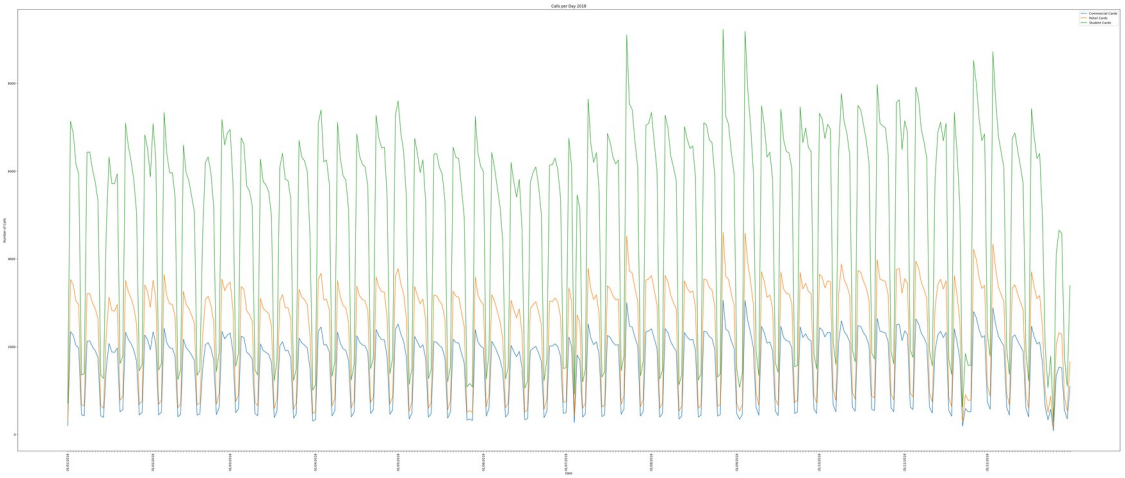


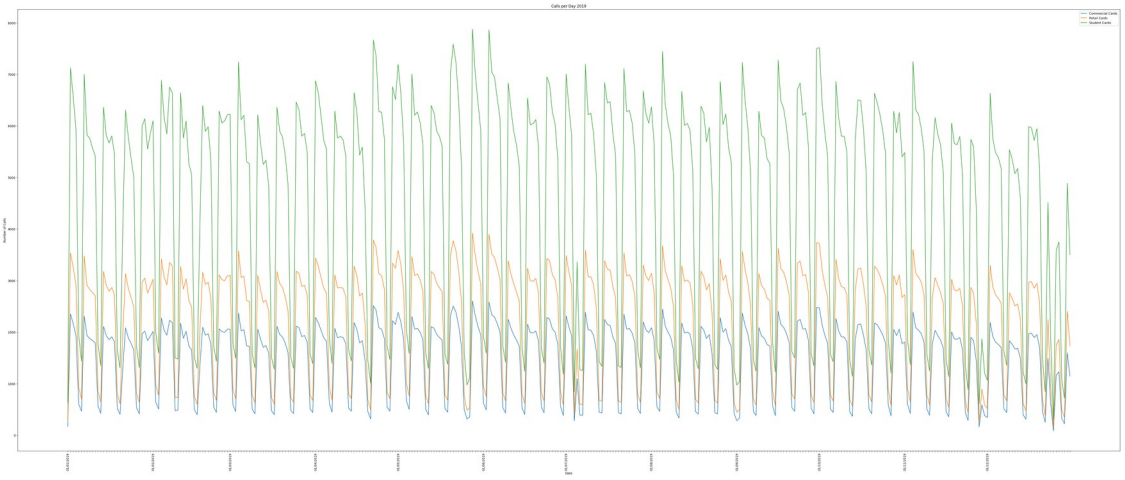


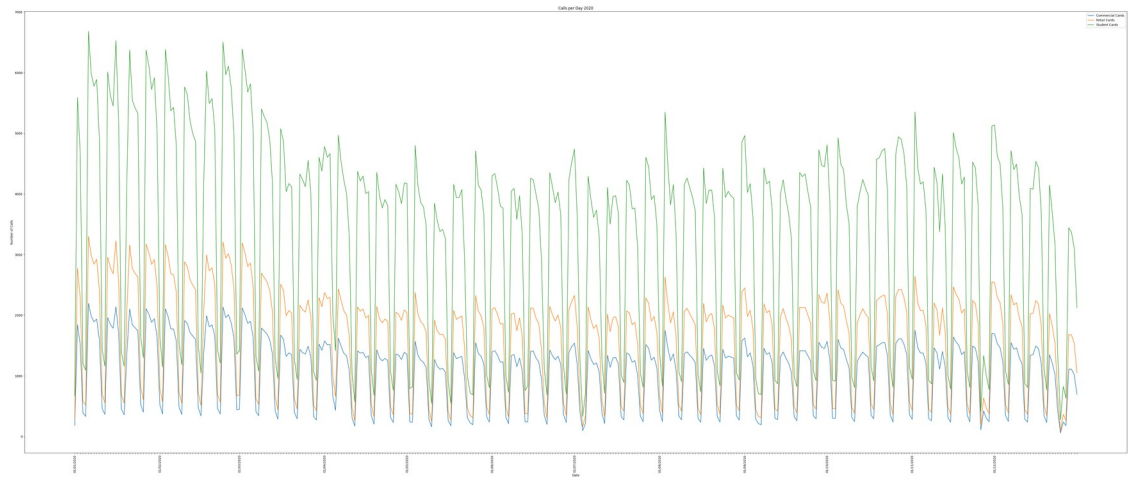


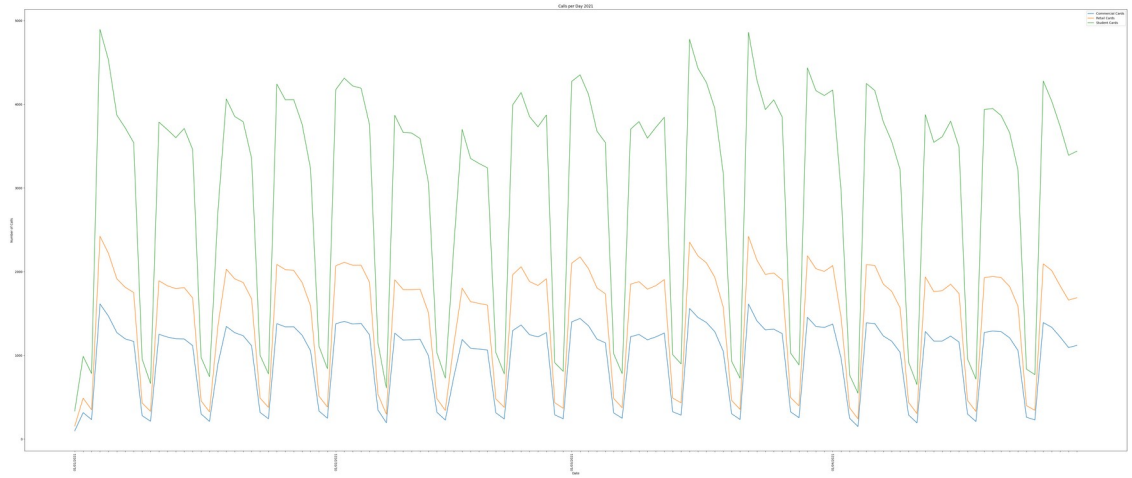
```
# Plot sorted data
years = df_not17_sorted.year.unique()
for y in years:
    x = range((df_not17_sorted['year'] == y).sum())
    cc_values = df_not17_sorted[df_not17_sorted.year == y].cc
    rc_values = df_not17_sorted[df_not17_sorted.year == y].rc
```

```
    sc_values = df_not17_sorted[df_not17_sorted.year == y].sc
    fig = plt.figure()
    fig.set_figheight(20)
    fig.set_figwidth(50)
    ax = fig.add_axes([1,1,1,1])
    ax.plot(x, cc_values, label='Commercial Cards')
    ax.plot(x, rc_values, label='Retail Cards')
    ax.plot(x, sc_values, label='Student Cards')
    ax.set_xticks(x, [ dt if arrow.get(dt, "DD/MM/YYYY").day==1 else
"" for dt in df_not17_sorted[df_not17_sorted.year == y]['date'] ],
rotation=90)
    ax.set_xlabel("Date")
    ax.set_ylabel("Number of Calls")
    ax.set_title(f"Calls per Day {y}")
    ax.legend()
    plt.show()
```



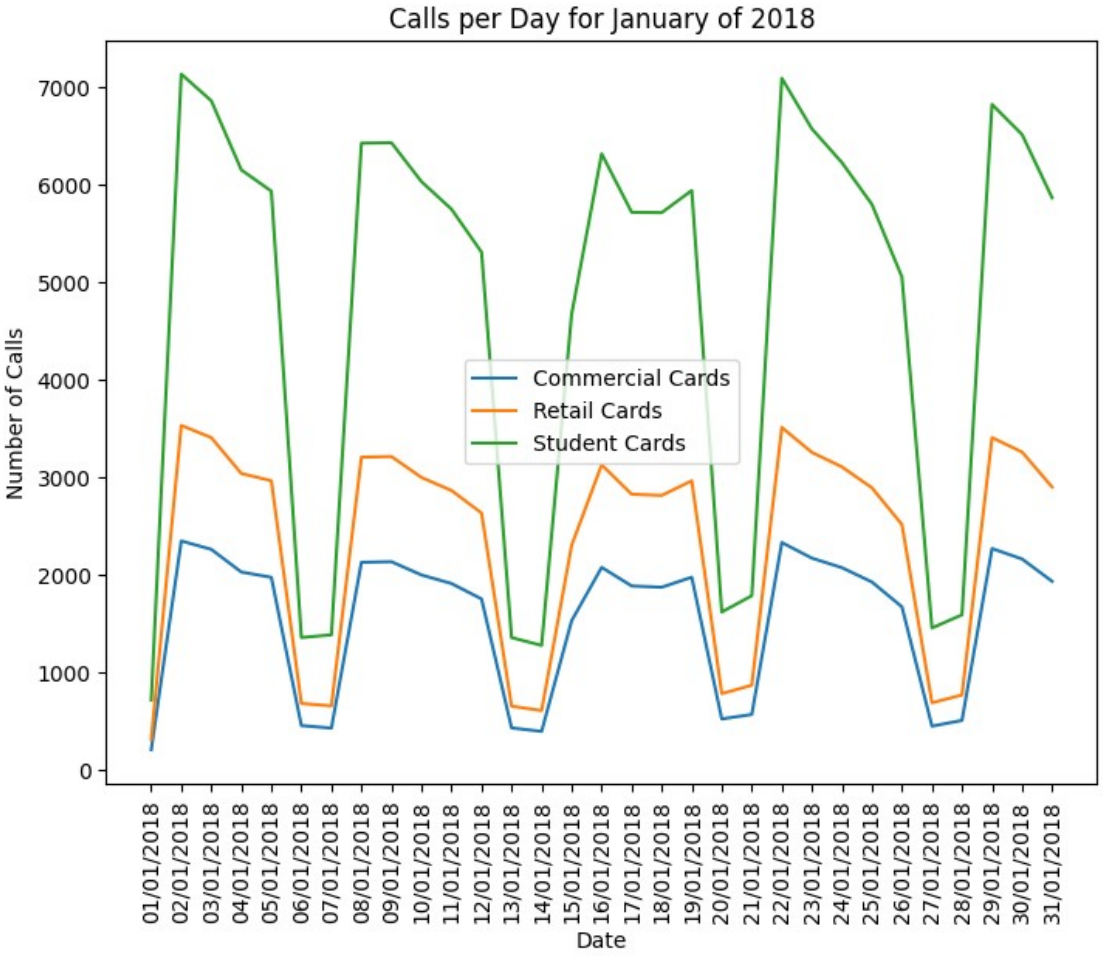


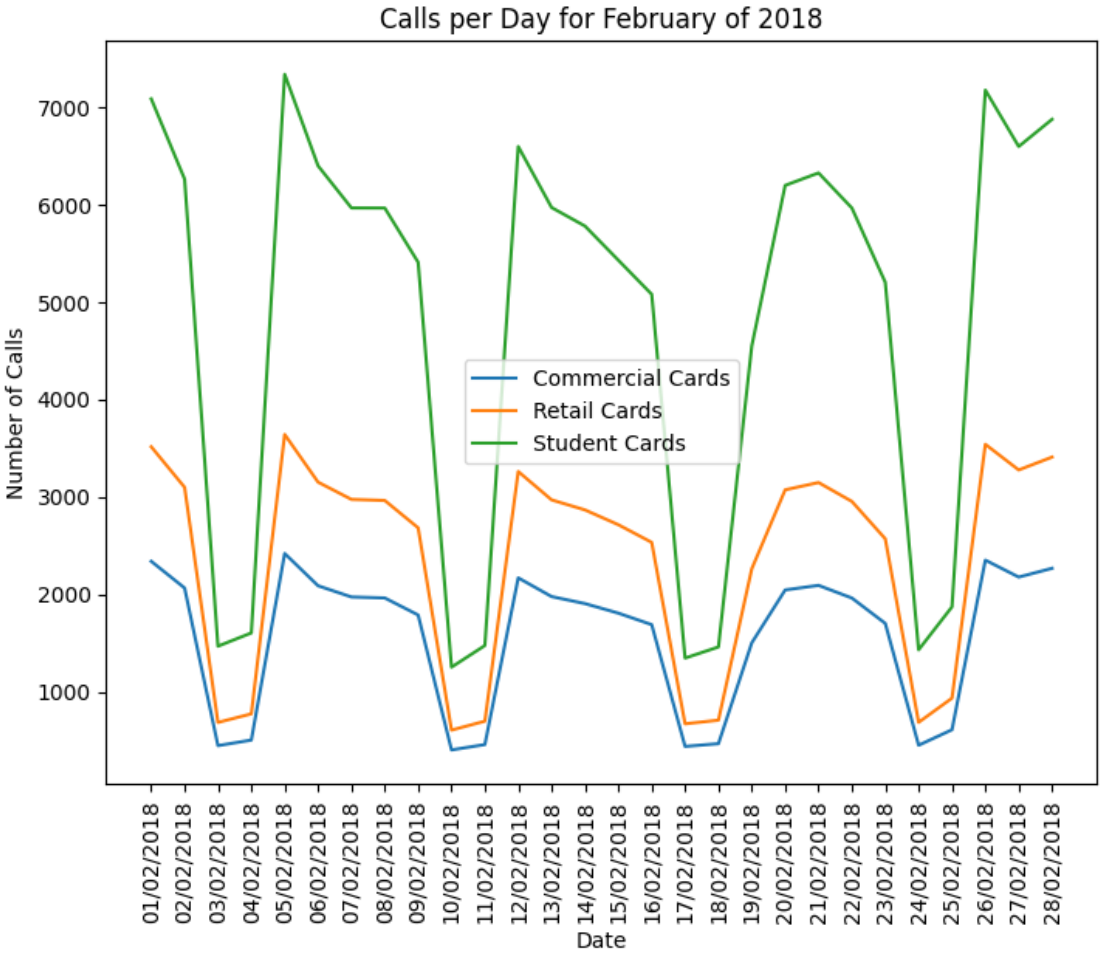


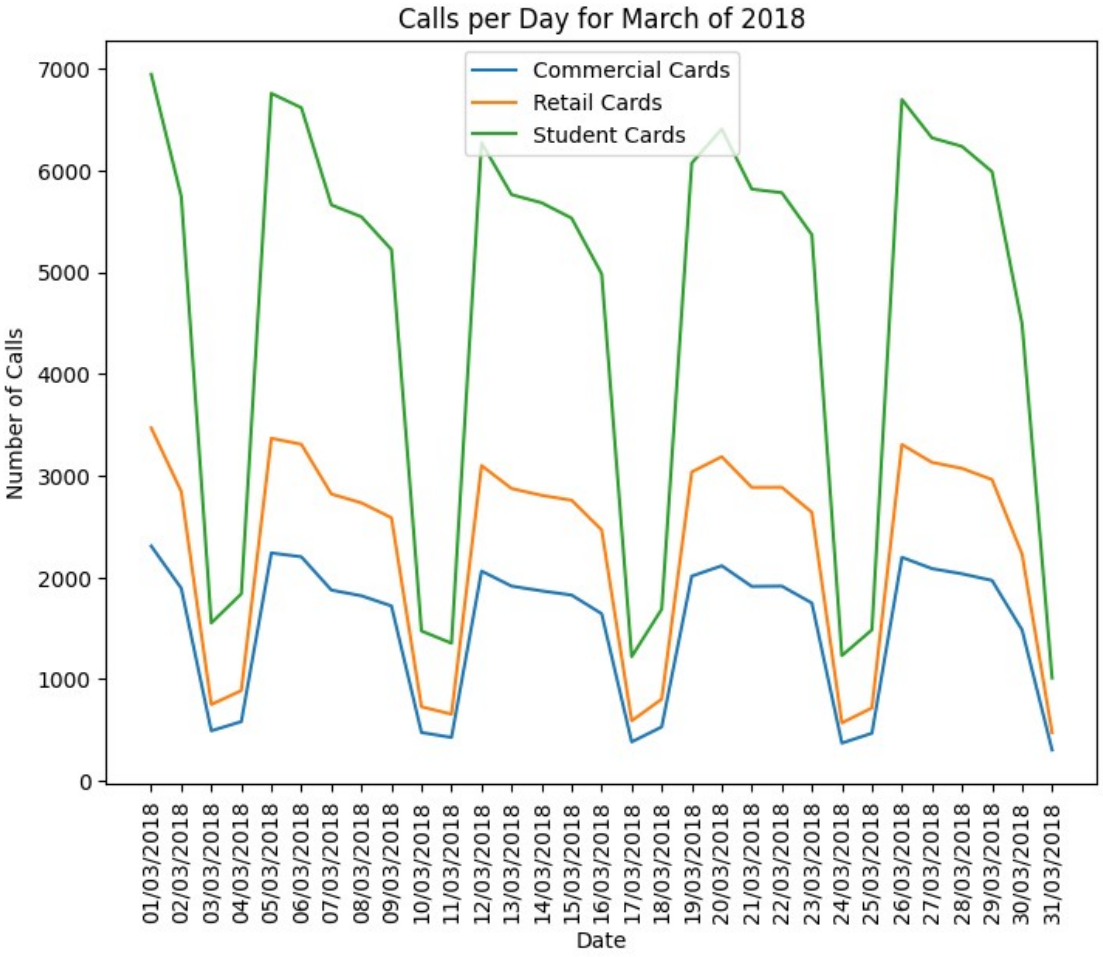


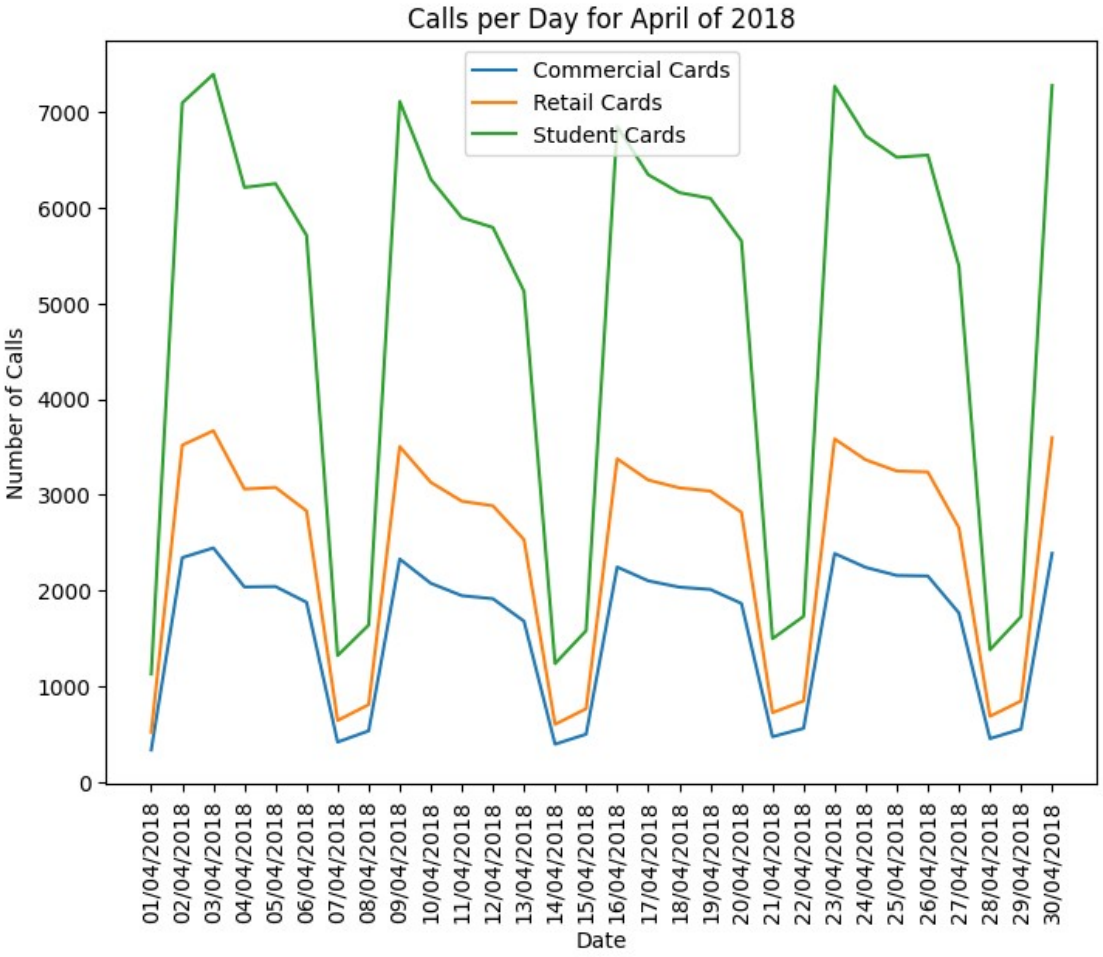
```
# Plot monthly sorted data for every year 2018-2021
years = df_not17_sorted.year.unique()
for y in years:
    df_not17_sorted_yearly = df_not17_sorted[df_not17_sorted.year ==
y]
    months = df_not17_sorted_yearly.month.unique()
    for m in months:
        x = range((df_not17_sorted_yearly.month == m).sum())
        cc_values =
df_not17_sorted_yearly[df_not17_sorted_yearly.month == m].cc
        rc_values =
df_not17_sorted_yearly[df_not17_sorted_yearly.month == m].rc
        sc_values =
df_not17_sorted_yearly[df_not17_sorted_yearly.month == m].sc
        fig = plt.figure()
        ax = fig.add_axes([1,1,1,1])
        ax.plot(x, cc_values, label='Commercial Cards')
        ax.plot(x, rc_values, label='Retail Cards')
        ax.plot(x, sc_values, label='Student Cards')
        ax.set_xticks(x,
df_not17_sorted_yearly[df_not17_sorted_yearly.month == m].date,
```

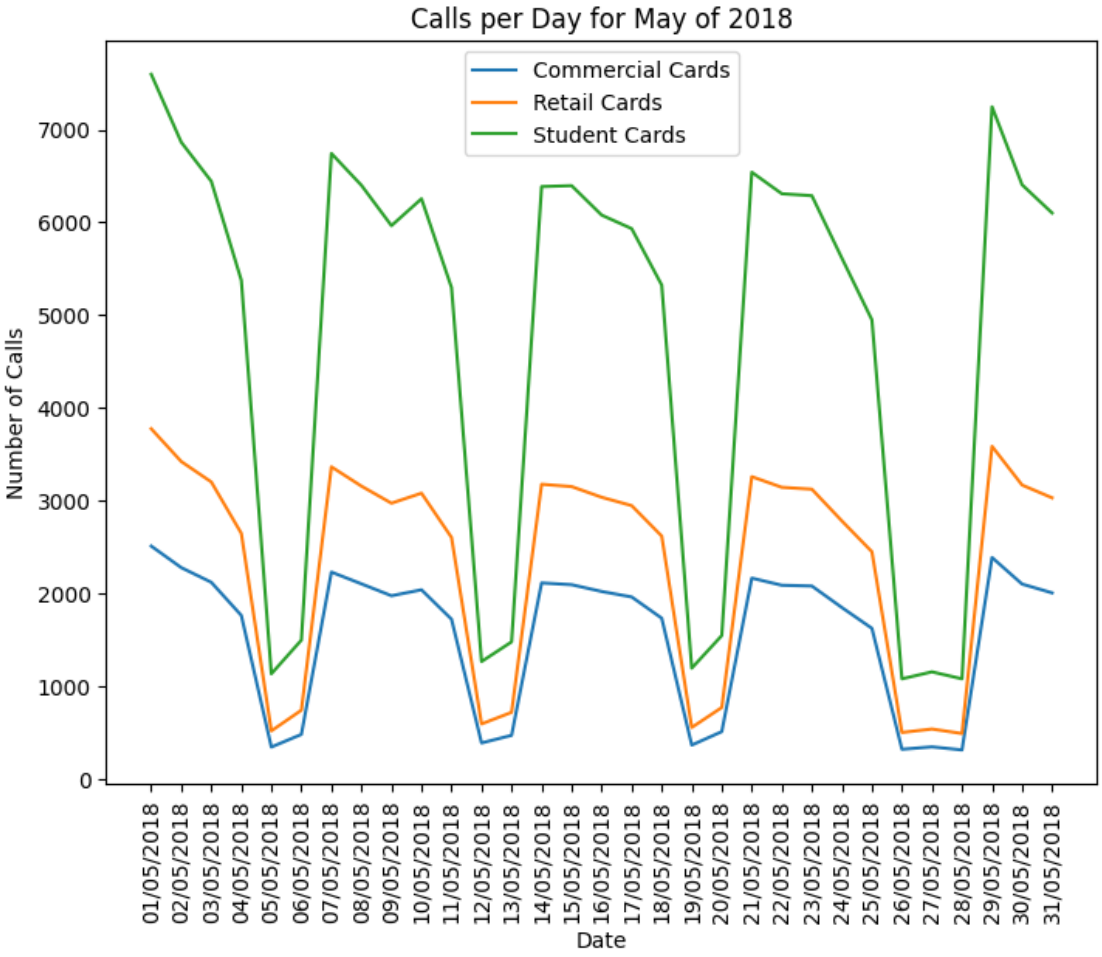
```
rotation=90)
        ax.set_xlabel("Date")
        ax.set_ylabel("Number of Calls")
        ax.set_title(f"Calls per Day for {arrow.get(str(m),
'M').format('MMMM')} of {y}")
        ax.legend()
        plt.show()
```

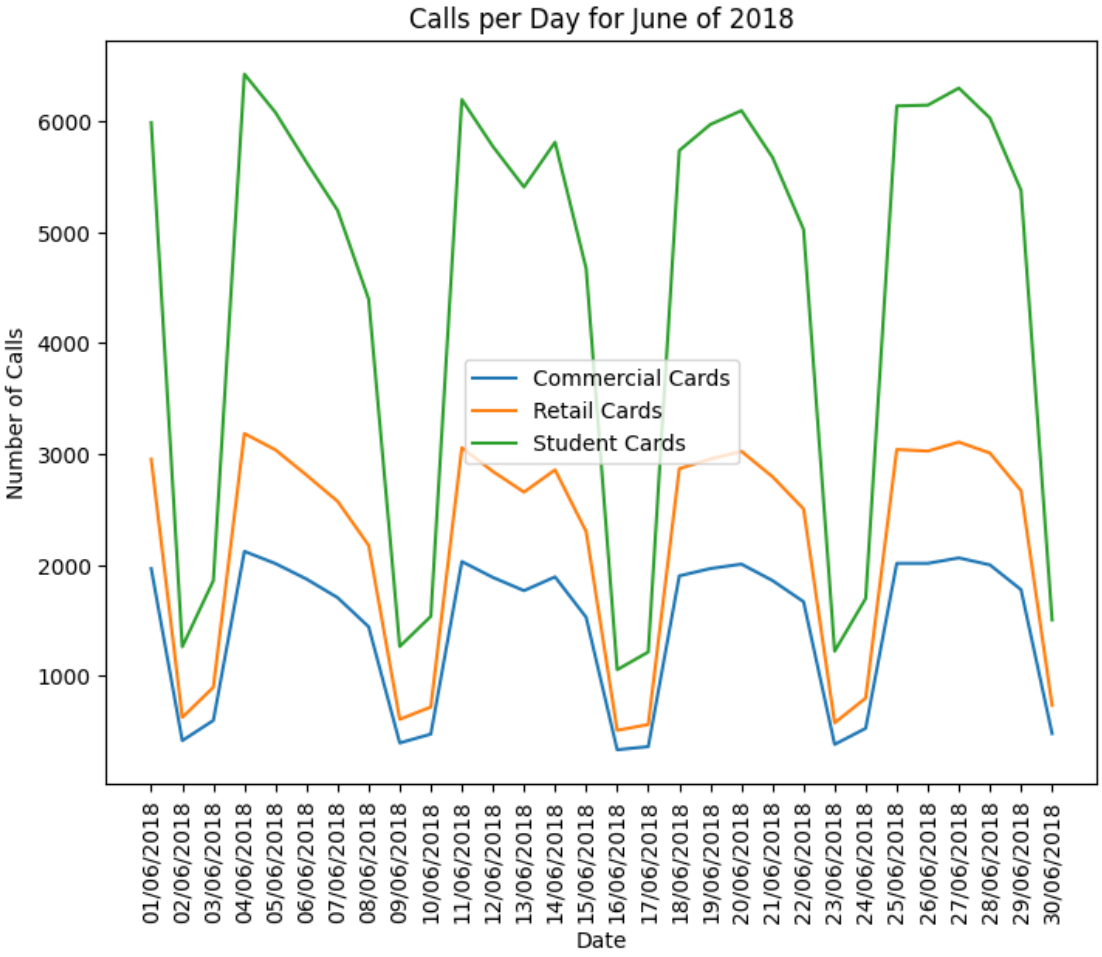


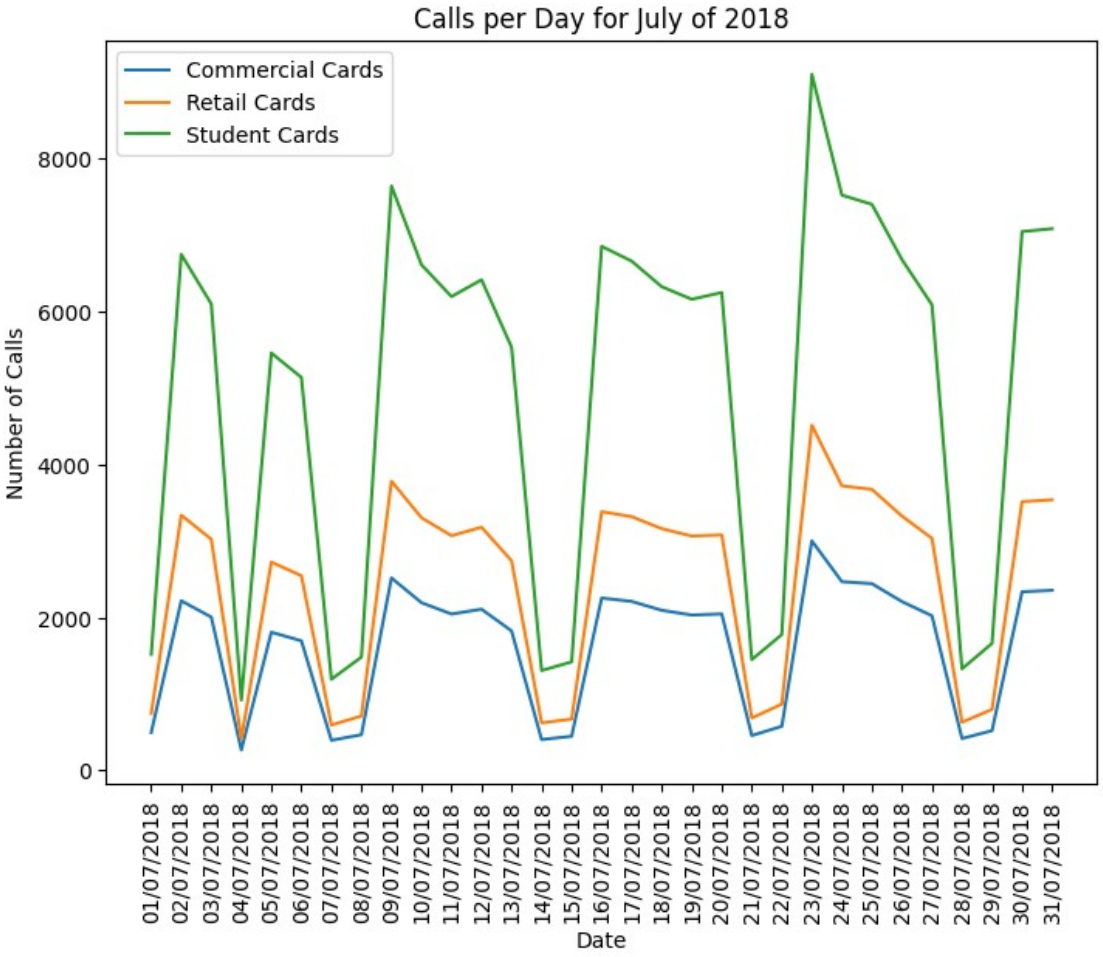

Calls per Day for February of 2018
Number of Calls
7000
6000
5000
4000
3000
2000
1000
Commercial Cards
Retail Cards
Student Cards
01/02/2018
02/02/2018
03/02/2018
04/02/2018
05/02/2018
06/02/2018
07/02/2018
08/02/2018
09/02/2018
10/02/2018
11/02/2018
12/02/2018
13/02/2018
14/02/2018
15/02/2018
16/02/2018
17/02/2018
18/02/2018
19/02/2018
20/02/2018
21/02/2018
22/02/2018
23/02/2018
24/02/2018
25/02/2018
26/02/2018
27/02/2018
28/02/2018
Date

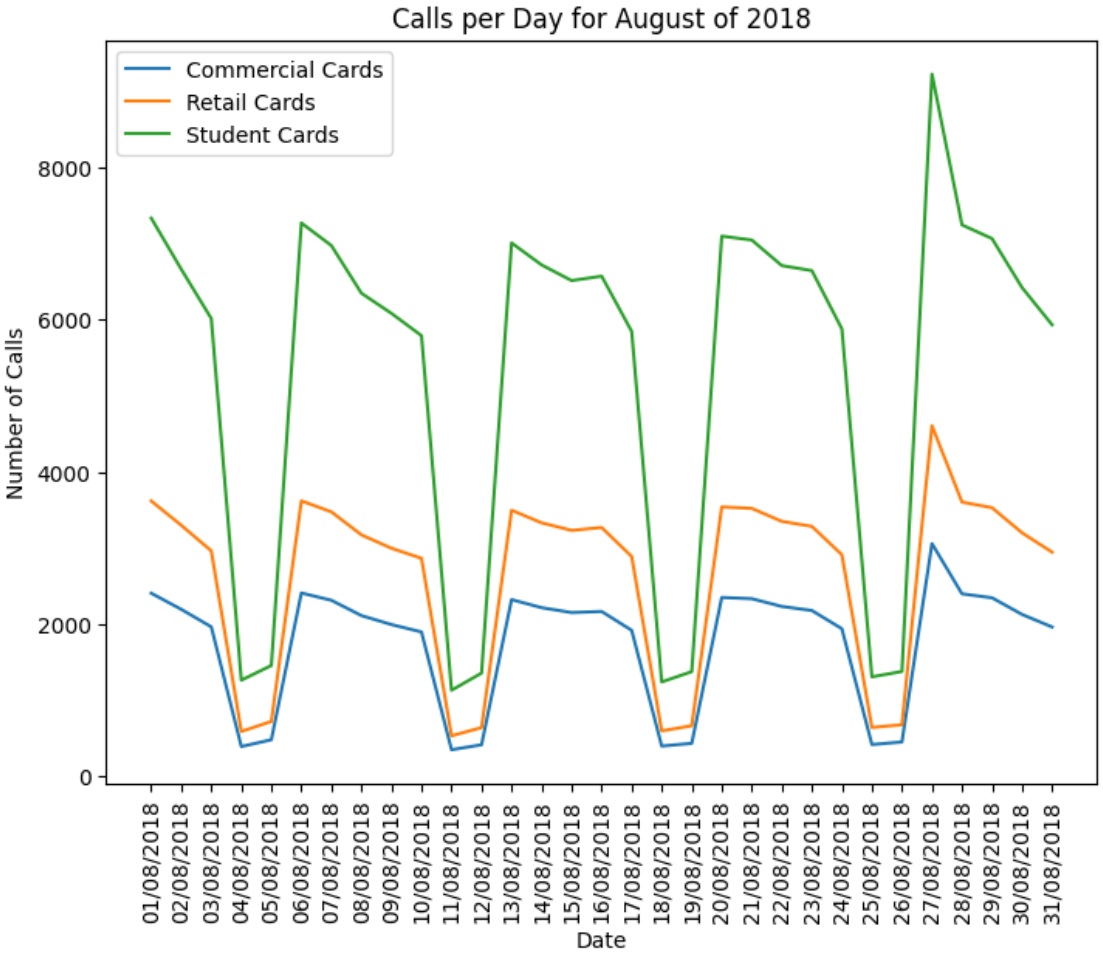

Calls per Day for March of 2018
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/03/2018
02/03/2018
03/03/2018
04/03/2018
05/03/2018
06/03/2018
07/03/2018
08/03/2018
09/03/2018
10/03/2018
11/03/2018
12/03/2018
13/03/2018
14/03/2018
15/03/2018
16/03/2018
17/03/2018
18/03/2018
19/03/2018
20/03/2018
21/03/2018
22/03/2018
23/03/2018
24/03/2018
25/03/2018
26/03/2018
27/03/2018
28/03/2018
29/03/2018
30/03/2018
31/03/2018
Date

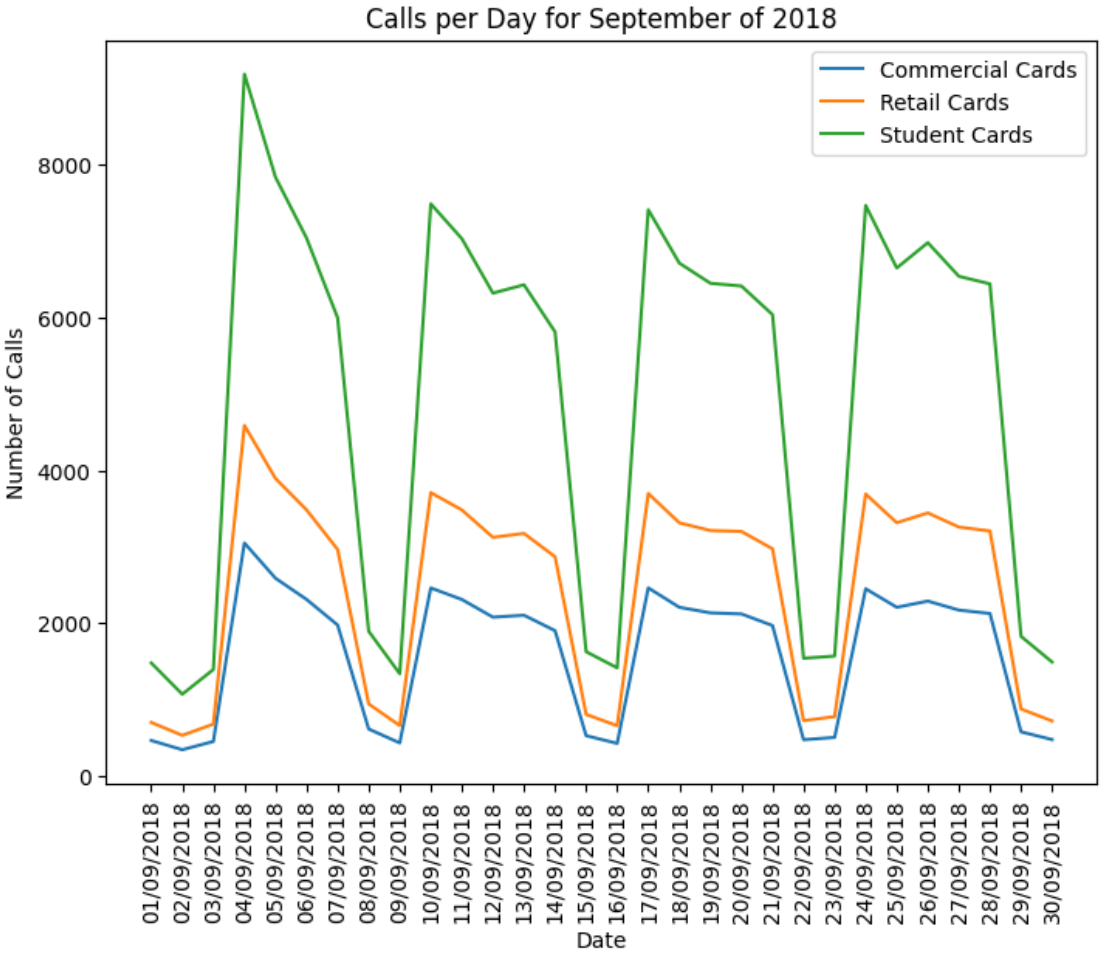

Calls per Day for April of 2018
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/04/2018
02/04/2018
03/04/2018
04/04/2018
05/04/2018
06/04/2018
07/04/2018
08/04/2018
09/04/2018
10/04/2018
11/04/2018
12/04/2018
13/04/2018
14/04/2018
15/04/2018
16/04/2018
17/04/2018
18/04/2018
19/04/2018
20/04/2018
21/04/2018
22/04/2018
23/04/2018
24/04/2018
25/04/2018
26/04/2018
27/04/2018
28/04/2018
29/04/2018
30/04/2018
Date

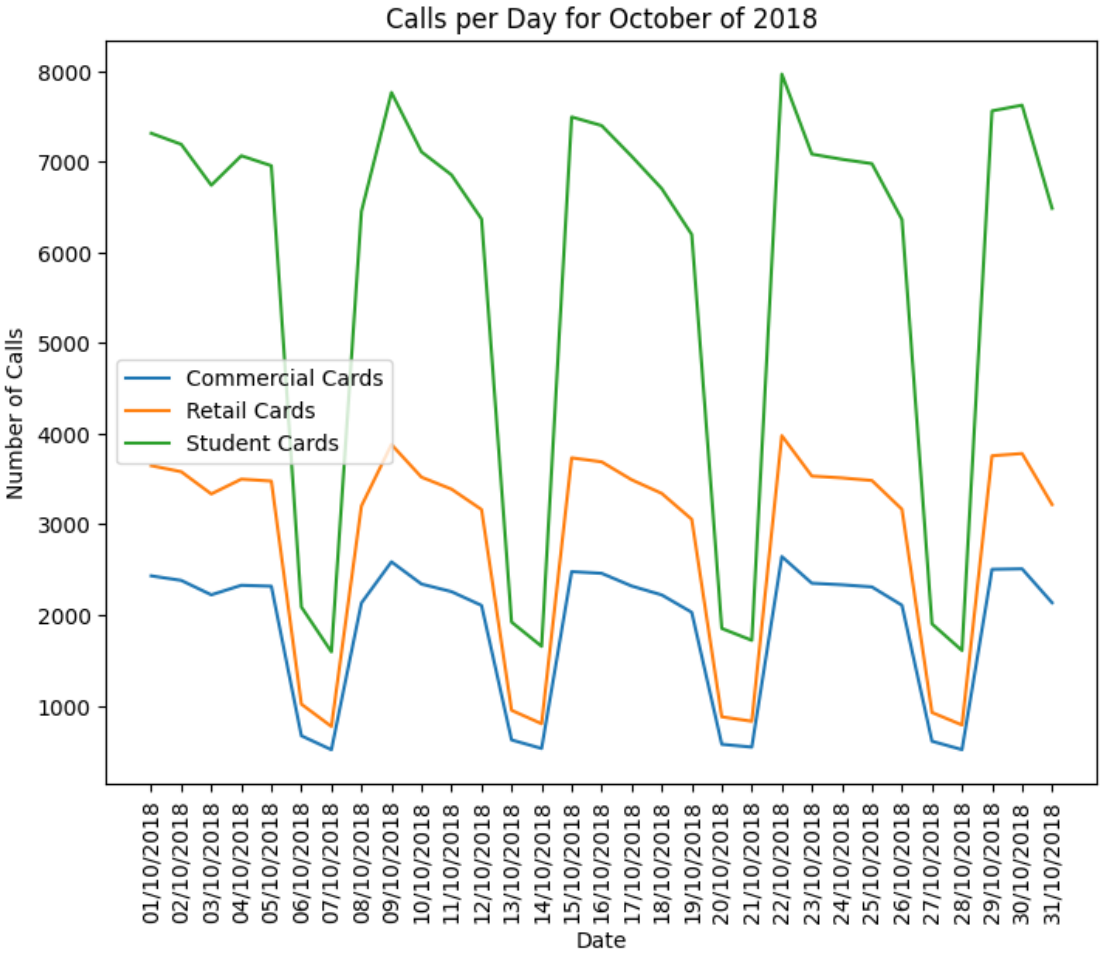

Calls per Day for May of 2018
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/05/2018
02/05/2018
03/05/2018
04/05/2018
05/05/2018
06/05/2018
07/05/2018
08/05/2018
09/05/2018
10/05/2018
11/05/2018
12/05/2018
13/05/2018
14/05/2018
15/05/2018
16/05/2018
17/05/2018
18/05/2018
19/05/2018
20/05/2018
21/05/2018
22/05/2018
23/05/2018
24/05/2018
25/05/2018
26/05/2018
27/05/2018
28/05/2018
29/05/2018
30/05/2018
31/05/2018
Date

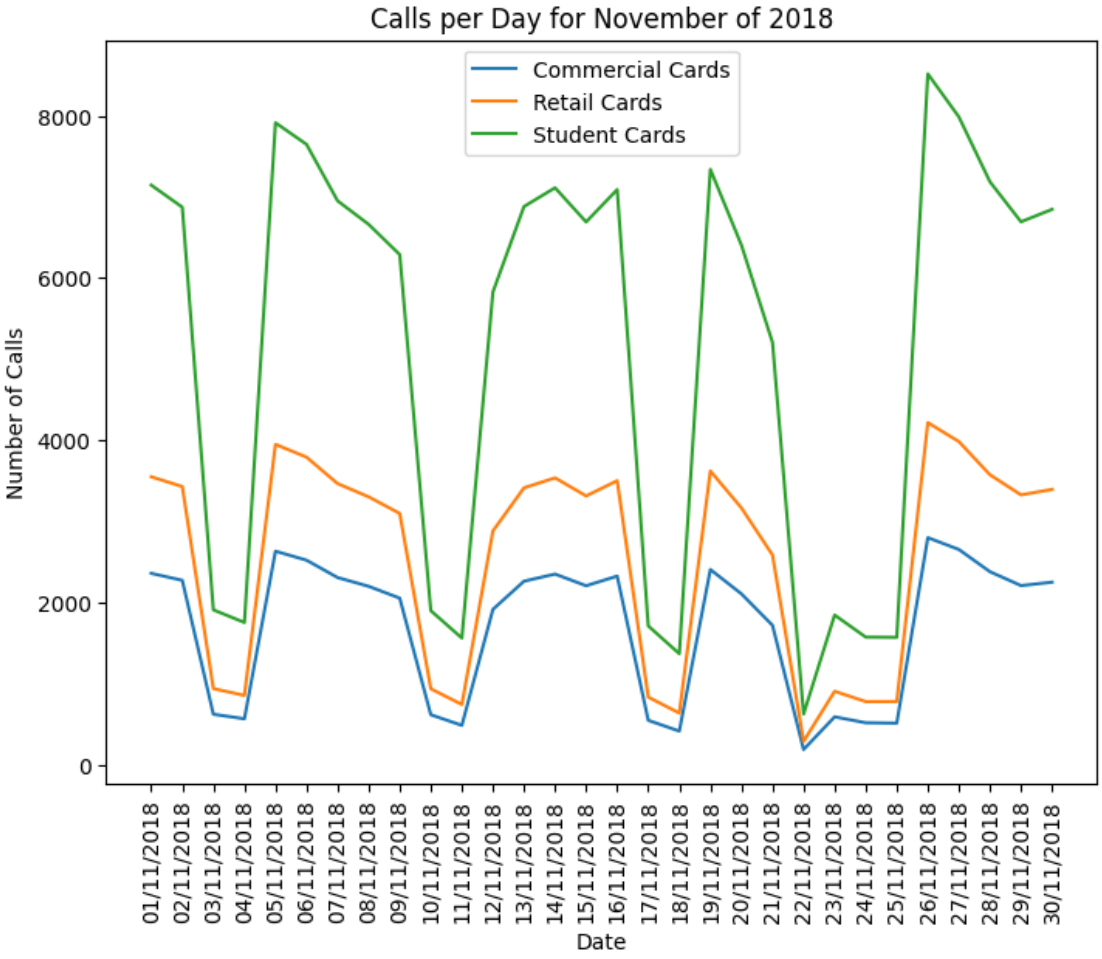

Calls per Day for June of 2018
Number of Calls
6000
5000
4000
3000
2000
1000
Commercial Cards
Retail Cards
Student Cards
01/06/2018
02/06/2018
03/06/2018
04/06/2018
05/06/2018
06/06/2018
07/06/2018
08/06/2018
09/06/2018
10/06/2018
11/06/2018
12/06/2018
13/06/2018
14/06/2018
15/06/2018
16/06/2018
17/06/2018
18/06/2018
19/06/2018
20/06/2018
21/06/2018
22/06/2018
23/06/2018
24/06/2018
25/06/2018
26/06/2018
27/06/2018
28/06/2018
29/06/2018
30/06/2018
Date

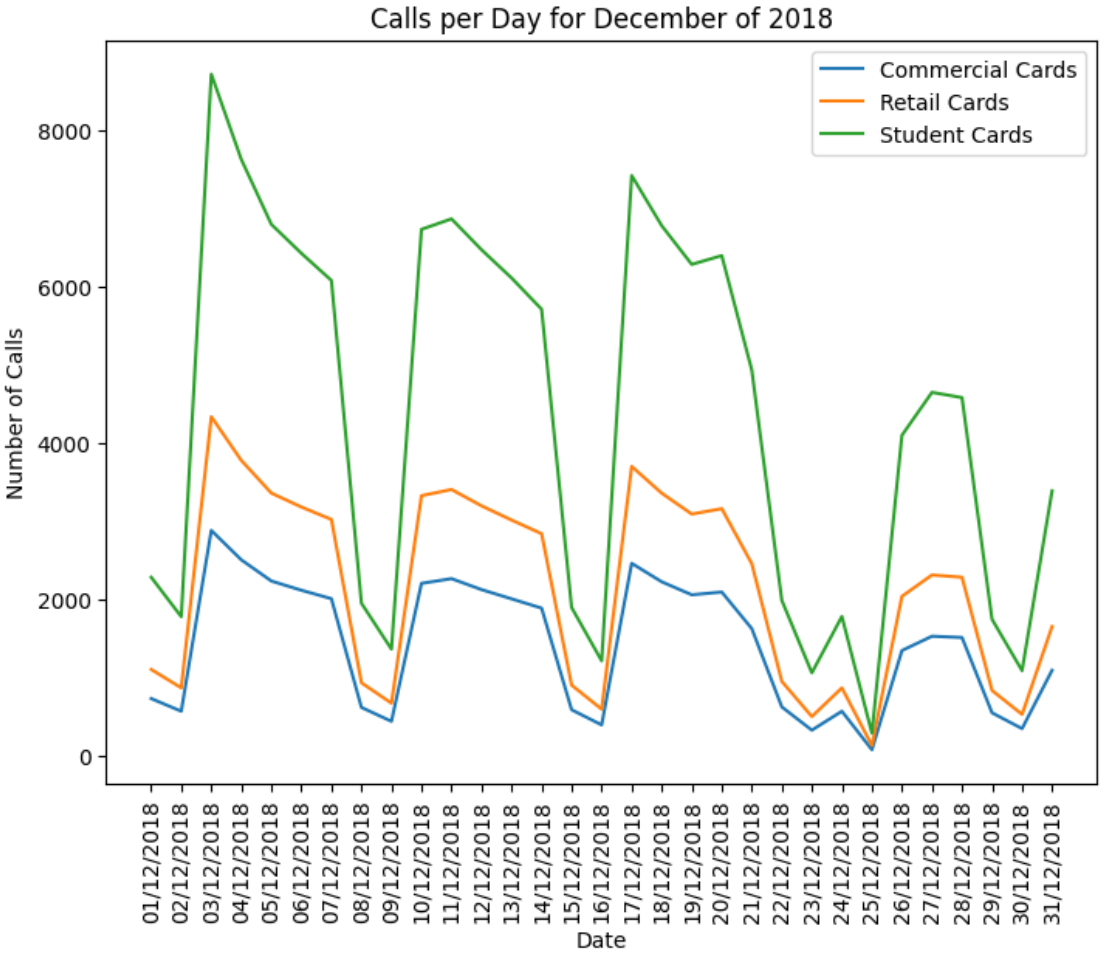

Calls per Day for July of 2018
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
2000
4000
6000
8000
01/07/2018
02/07/2018
03/07/2018
04/07/2018
05/07/2018
06/07/2018
07/07/2018
08/07/2018
09/07/2018
10/07/2018
11/07/2018
12/07/2018
13/07/2018
14/07/2018
15/07/2018
16/07/2018
17/07/2018
18/07/2018
19/07/2018
20/07/2018
21/07/2018
22/07/2018
23/07/2018
24/07/2018
25/07/2018
26/07/2018
27/07/2018
28/07/2018
29/07/2018
30/07/2018
31/07/2018
Date

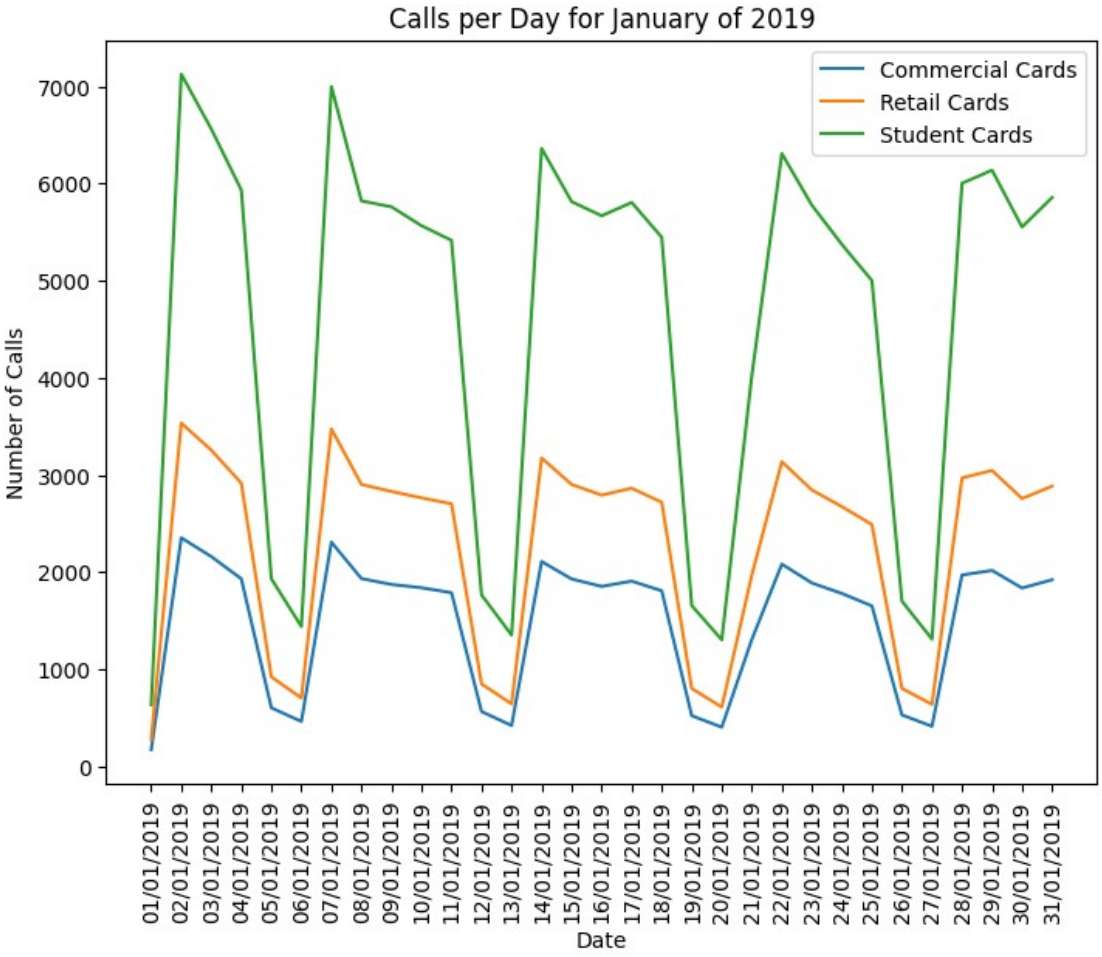

Calls per Day for August of 2018
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
2000
4000
6000
8000
01/08/2018
02/08/2018
03/08/2018
04/08/2018
05/08/2018
06/08/2018
07/08/2018
08/08/2018
09/08/2018
10/08/2018
11/08/2018
12/08/2018
13/08/2018
14/08/2018
15/08/2018
16/08/2018
17/08/2018
18/08/2018
19/08/2018
20/08/2018
21/08/2018
22/08/2018
23/08/2018
24/08/2018
25/08/2018
26/08/2018
27/08/2018
28/08/2018
29/08/2018
30/08/2018
31/08/2018
Date

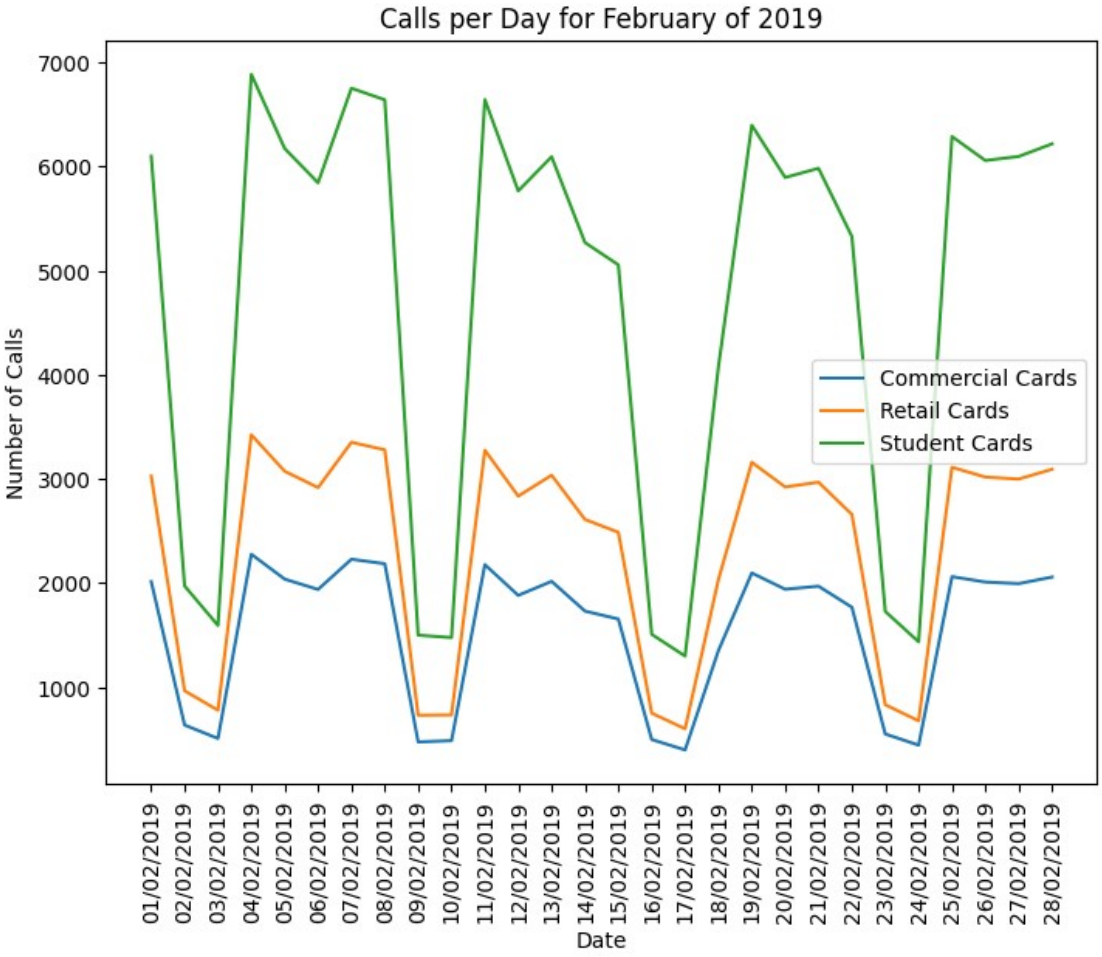

Calls per Day for September of 2018
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
2000
4000
6000
8000
01/09/2018
02/09/2018
03/09/2018
04/09/2018
05/09/2018
06/09/2018
07/09/2018
08/09/2018
09/09/2018
10/09/2018
11/09/2018
12/09/2018
13/09/2018
14/09/2018
15/09/2018
16/09/2018
17/09/2018
18/09/2018
19/09/2018
20/09/2018
21/09/2018
22/09/2018
23/09/2018
24/09/2018
25/09/2018
26/09/2018
27/09/2018
28/09/2018
29/09/2018
30/09/2018
Date

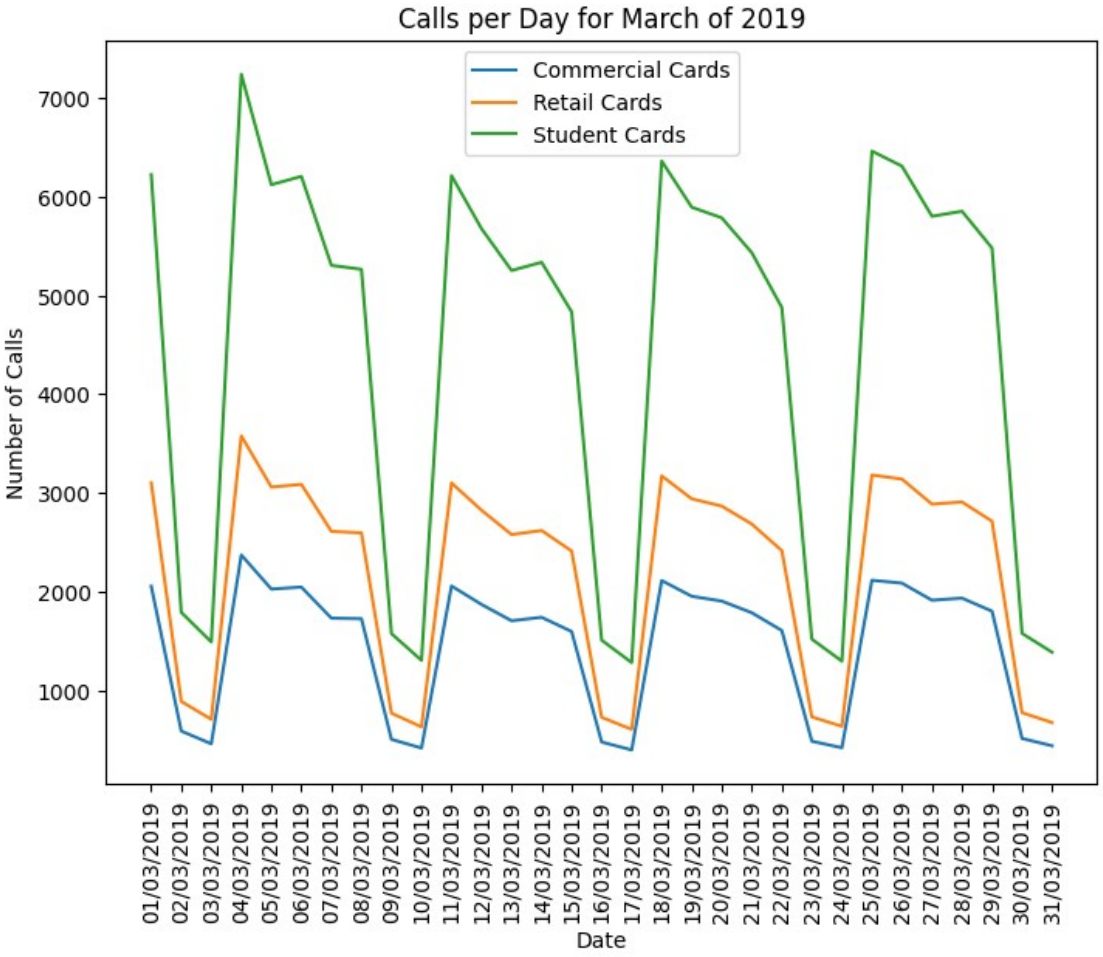

Calls per Day for October of 2018
Number of Calls
8000
7000
6000
5000
4000
3000
2000
1000
Commercial Cards
Retail Cards
Student Cards
01/10/2018
02/10/2018
03/10/2018
04/10/2018
05/10/2018
06/10/2018
07/10/2018
08/10/2018
09/10/2018
10/10/2018
11/10/2018
12/10/2018
13/10/2018
14/10/2018
15/10/2018
16/10/2018
17/10/2018
18/10/2018
19/10/2018
20/10/2018
21/10/2018
22/10/2018
23/10/2018
24/10/2018
25/10/2018
26/10/2018
27/10/2018
28/10/2018
29/10/2018
30/10/2018
31/10/2018
Date

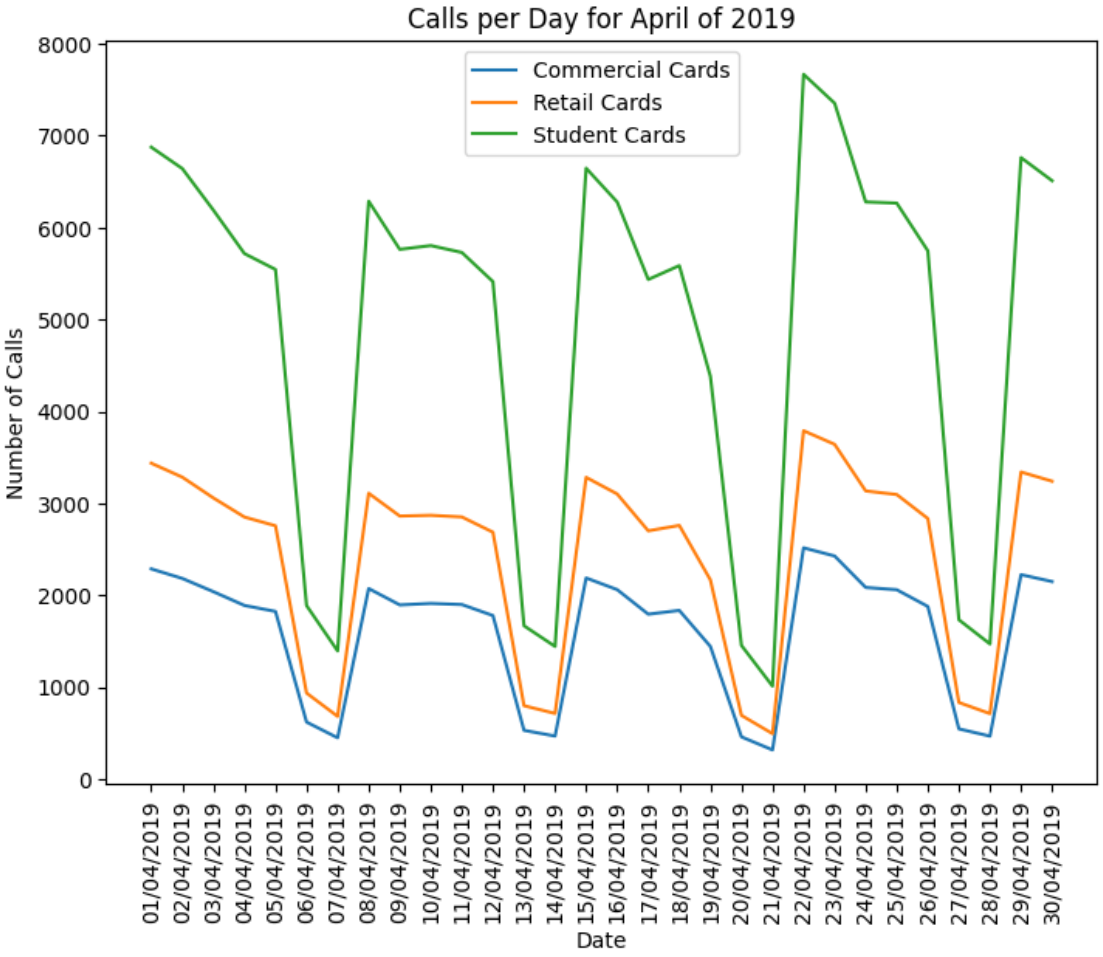

Calls per Day for November of 2018
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
2000
4000
6000
8000
01/11/2018
02/11/2018
03/11/2018
04/11/2018
05/11/2018
06/11/2018
07/11/2018
08/11/2018
09/11/2018
10/11/2018
11/11/2018
12/11/2018
13/11/2018
14/11/2018
15/11/2018
16/11/2018
17/11/2018
18/11/2018
19/11/2018
20/11/2018
21/11/2018
22/11/2018
23/11/2018
24/11/2018
25/11/2018
26/11/2018
27/11/2018
28/11/2018
29/11/2018
30/11/2018
Date

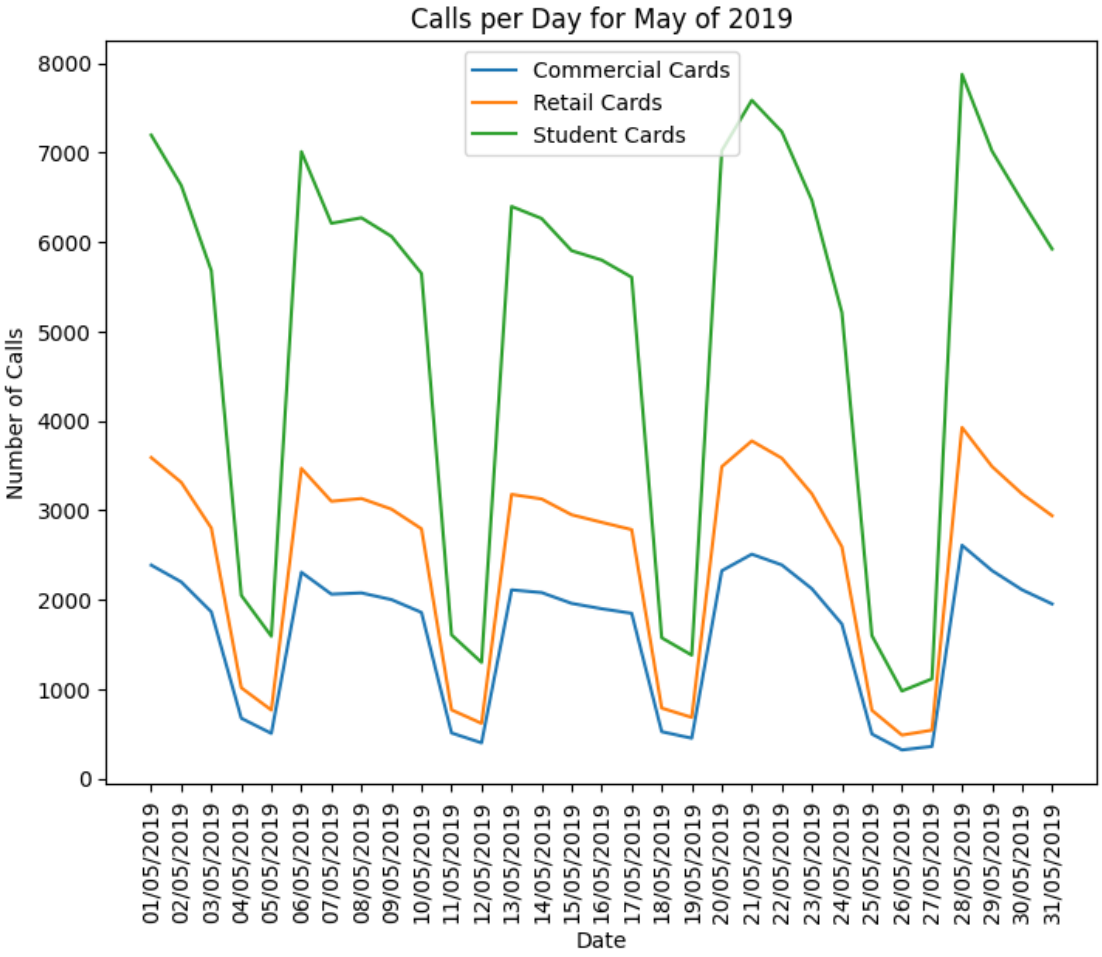

Calls per Day for December of 2018
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
2000
4000
6000
8000
01/12/2018
02/12/2018
03/12/2018
04/12/2018
05/12/2018
06/12/2018
07/12/2018
08/12/2018
09/12/2018
10/12/2018
11/12/2018
12/12/2018
13/12/2018
14/12/2018
15/12/2018
16/12/2018
17/12/2018
18/12/2018
19/12/2018
20/12/2018
21/12/2018
22/12/2018
23/12/2018
24/12/2018
25/12/2018
26/12/2018
27/12/2018
28/12/2018
29/12/2018
30/12/2018
31/12/2018
Date

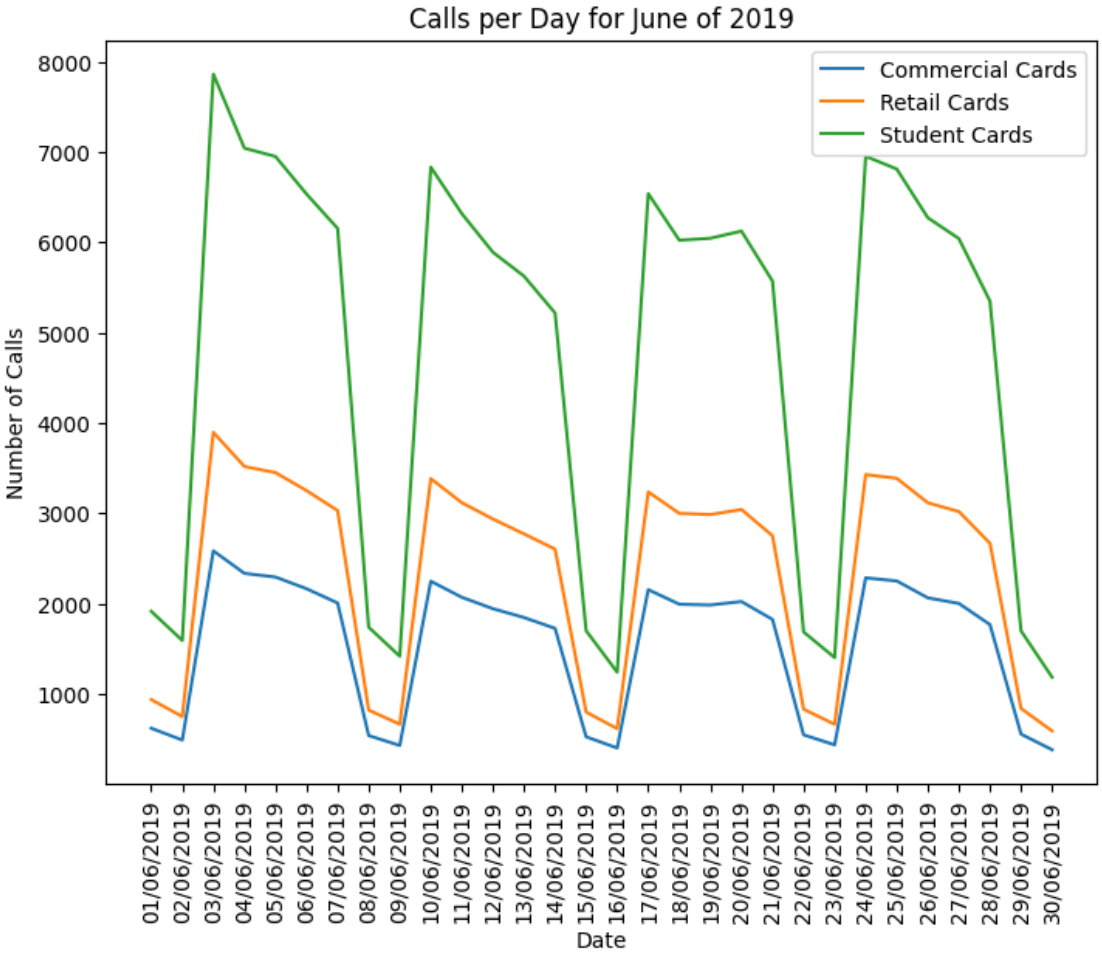

Calls per Day for January of 2019
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/01/2019
02/01/2019
03/01/2019
04/01/2019
05/01/2019
06/01/2019
07/01/2019
08/01/2019
09/01/2019
10/01/2019
11/01/2019
12/01/2019
13/01/2019
14/01/2019
15/01/2019
16/01/2019
17/01/2019
18/01/2019
19/01/2019
20/01/2019
21/01/2019
22/01/2019
23/01/2019
24/01/2019
25/01/2019
26/01/2019
27/01/2019
28/01/2019
29/01/2019
30/01/2019
31/01/2019
Date

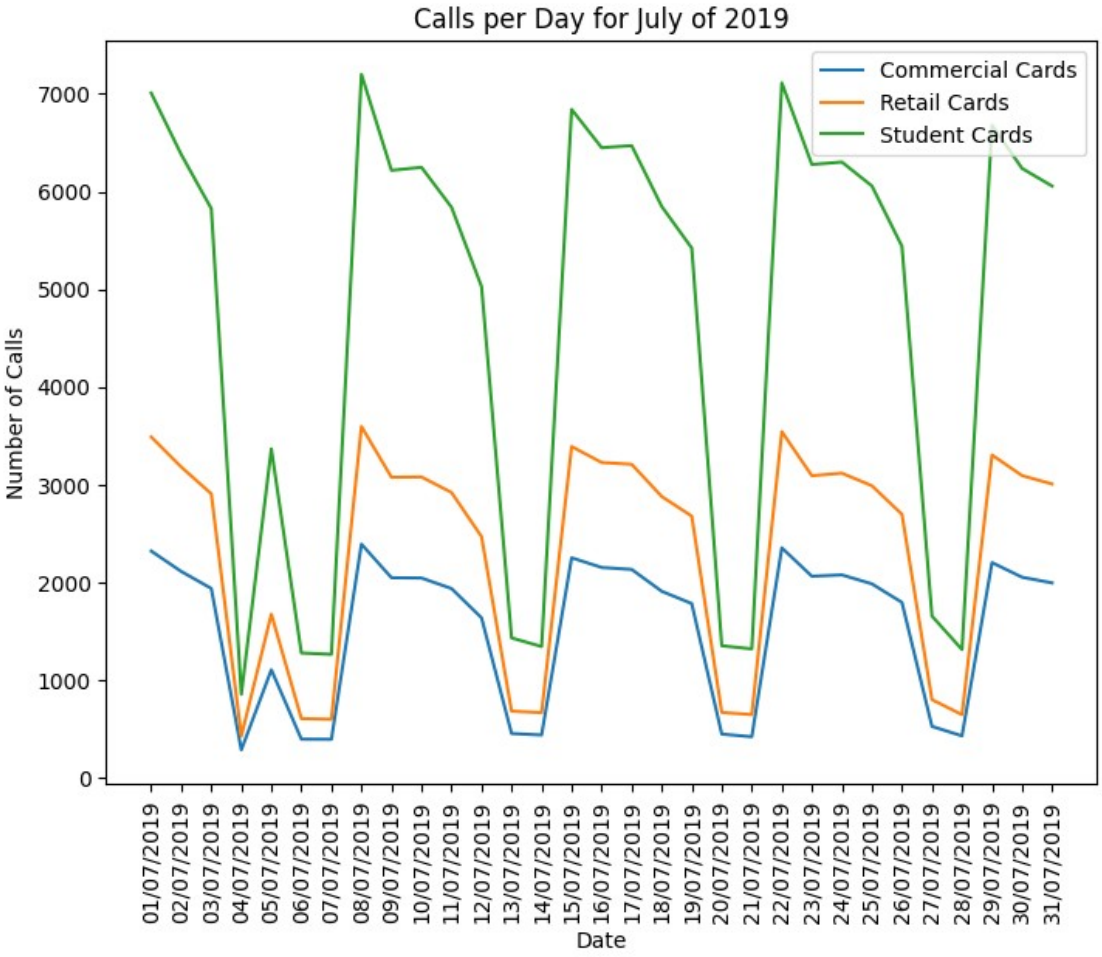

Calls per Day for February of 2019
Number of Calls
7000
6000
5000
4000
3000
2000
1000
Commercial Cards
Retail Cards
Student Cards
01/02/2019
02/02/2019
03/02/2019
04/02/2019
05/02/2019
06/02/2019
07/02/2019
08/02/2019
09/02/2019
10/02/2019
11/02/2019
12/02/2019
13/02/2019
14/02/2019
15/02/2019
16/02/2019
17/02/2019
18/02/2019
19/02/2019
20/02/2019
21/02/2019
22/02/2019
23/02/2019
24/02/2019
25/02/2019
26/02/2019
27/02/2019
28/02/2019
Date

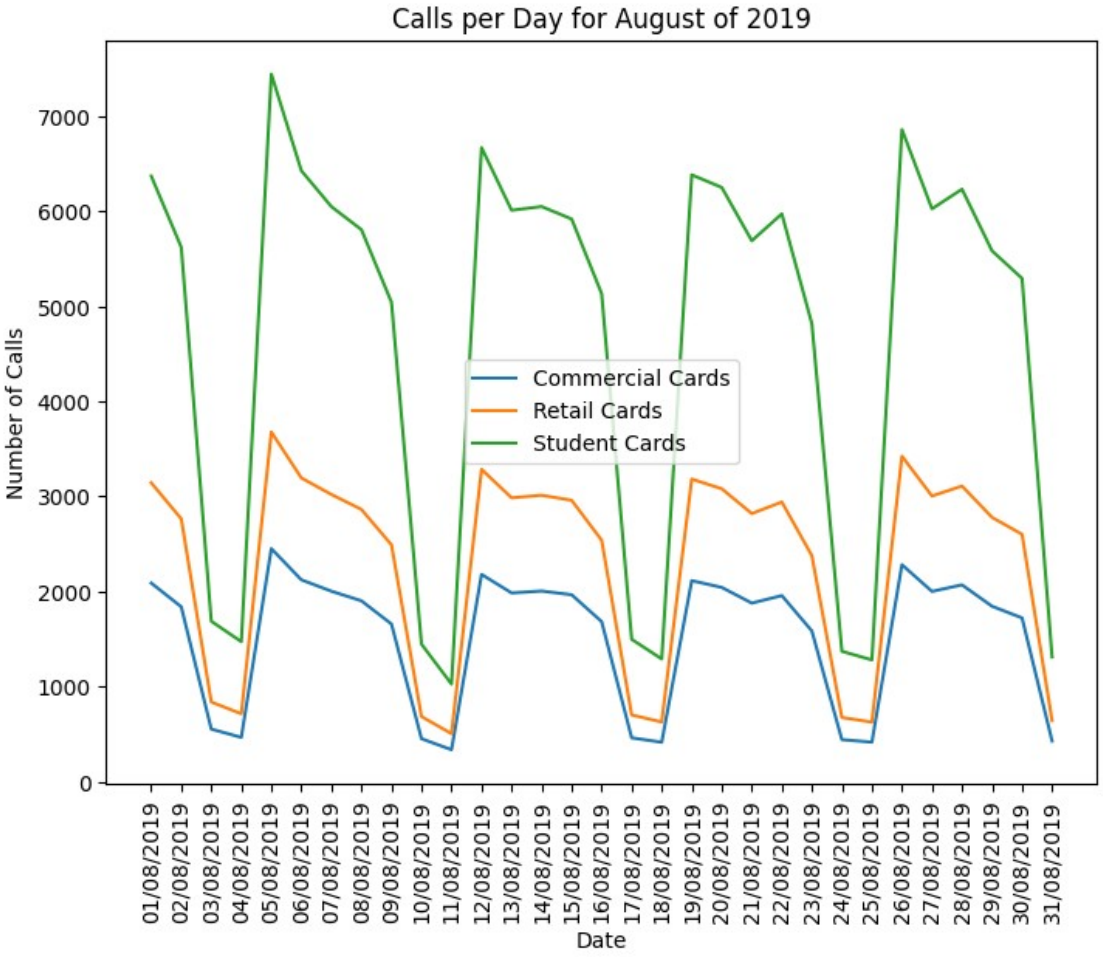

Calls per Day for March of 2019
Commercial Cards
Retail Cards
Student Cards
Number of Calls
7000
6000
5000
4000
3000
2000
1000
01/03/2019
02/03/2019
03/03/2019
04/03/2019
05/03/2019
06/03/2019
07/03/2019
08/03/2019
09/03/2019
10/03/2019
11/03/2019
12/03/2019
13/03/2019
14/03/2019
15/03/2019
16/03/2019
17/03/2019
18/03/2019
19/03/2019
20/03/2019
21/03/2019
22/03/2019
23/03/2019
24/03/2019
25/03/2019
26/03/2019
27/03/2019
28/03/2019
29/03/2019
30/03/2019
31/03/2019
Date

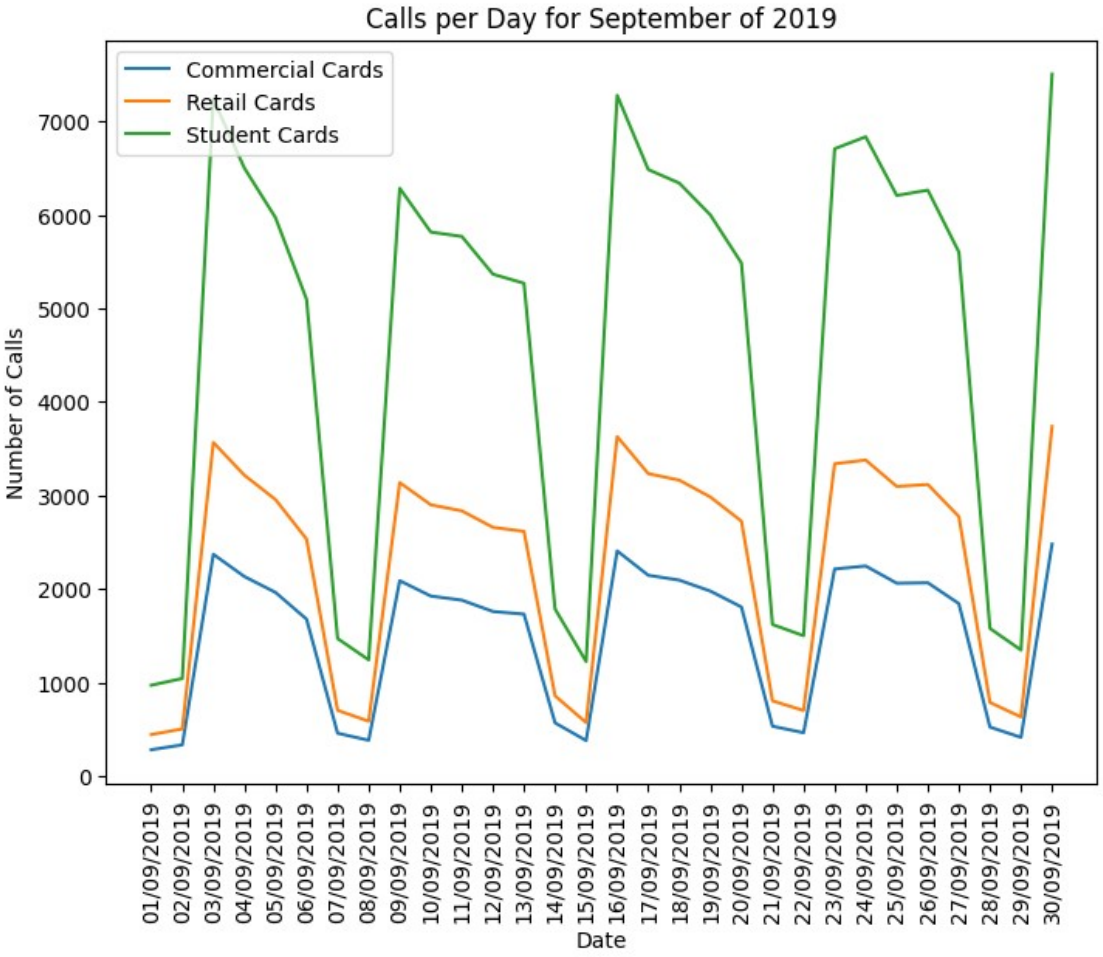

Calls per Day for April of 2019
Commercial Cards
Retail Cards
Student Cards
Number of Calls
8000
7000
6000
5000
4000
3000
2000
1000
0
01/04/2019
02/04/2019
03/04/2019
04/04/2019
05/04/2019
06/04/2019
07/04/2019
08/04/2019
09/04/2019
10/04/2019
11/04/2019
12/04/2019
13/04/2019
14/04/2019
15/04/2019
16/04/2019
17/04/2019
18/04/2019
19/04/2019
20/04/2019
21/04/2019
22/04/2019
23/04/2019
24/04/2019
25/04/2019
26/04/2019
27/04/2019
28/04/2019
29/04/2019
30/04/2019
Date

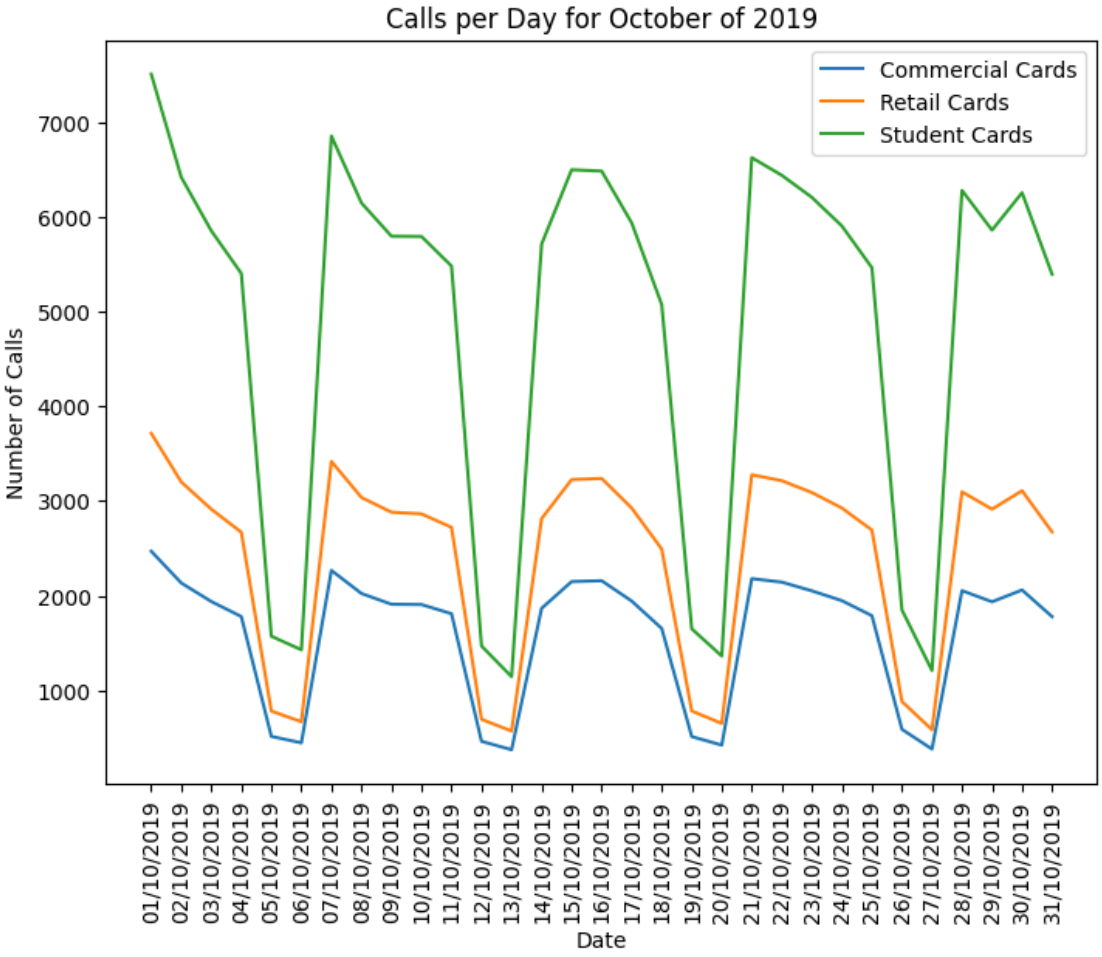

Calls per Day for May of 2019
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
8000
01/05/2019
02/05/2019
03/05/2019
04/05/2019
05/05/2019
06/05/2019
07/05/2019
08/05/2019
09/05/2019
10/05/2019
11/05/2019
12/05/2019
13/05/2019
14/05/2019
15/05/2019
16/05/2019
17/05/2019
18/05/2019
19/05/2019
20/05/2019
21/05/2019
22/05/2019
23/05/2019
24/05/2019
25/05/2019
26/05/2019
27/05/2019
28/05/2019
29/05/2019
30/05/2019
31/05/2019
Date

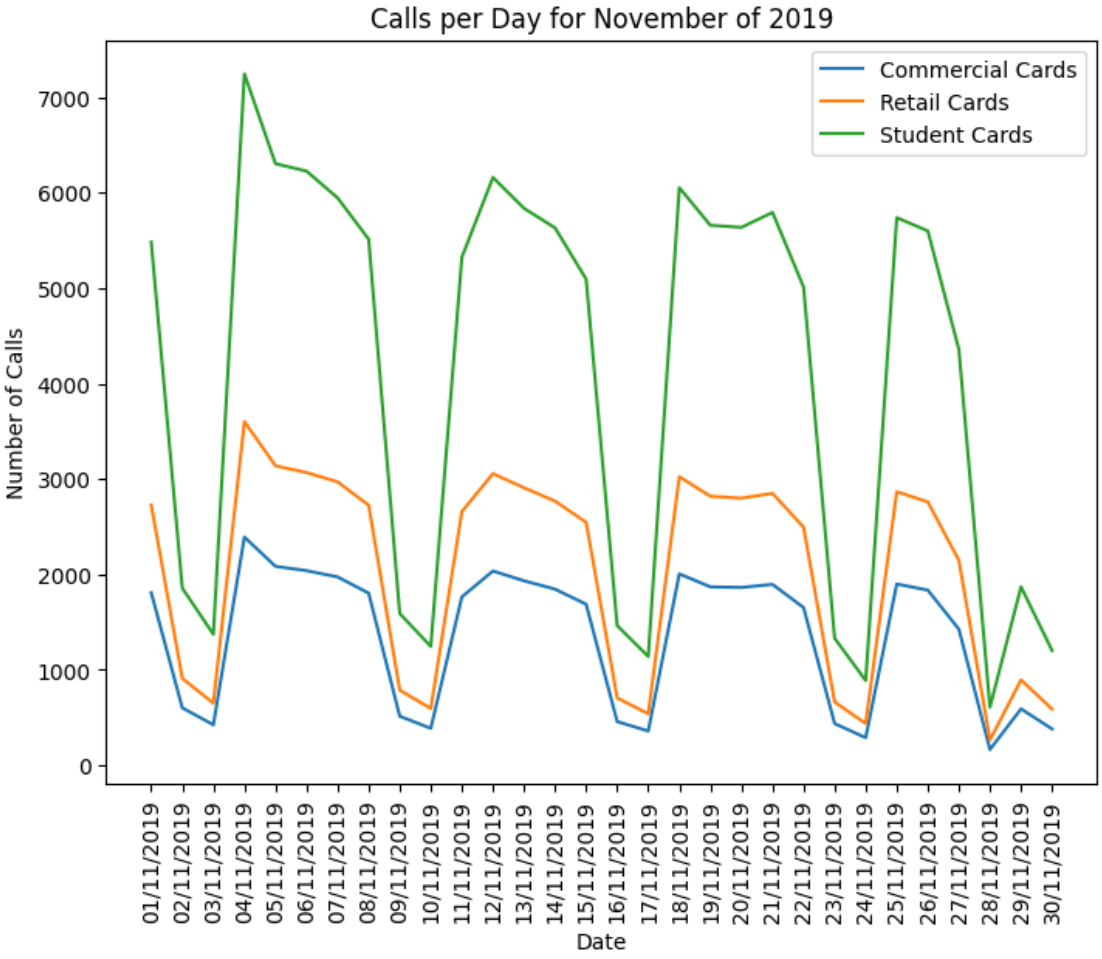

Calls per Day for June of 2019
Commercial Cards
Retail Cards
Student Cards
Number of Calls
8000
7000
6000
5000
4000
3000
2000
1000
01/06/2019
02/06/2019
03/06/2019
04/06/2019
05/06/2019
06/06/2019
07/06/2019
08/06/2019
09/06/2019
10/06/2019
11/06/2019
12/06/2019
13/06/2019
14/06/2019
15/06/2019
16/06/2019
17/06/2019
18/06/2019
19/06/2019
20/06/2019
21/06/2019
22/06/2019
23/06/2019
24/06/2019
25/06/2019
26/06/2019
27/06/2019
28/06/2019
29/06/2019
30/06/2019
Date

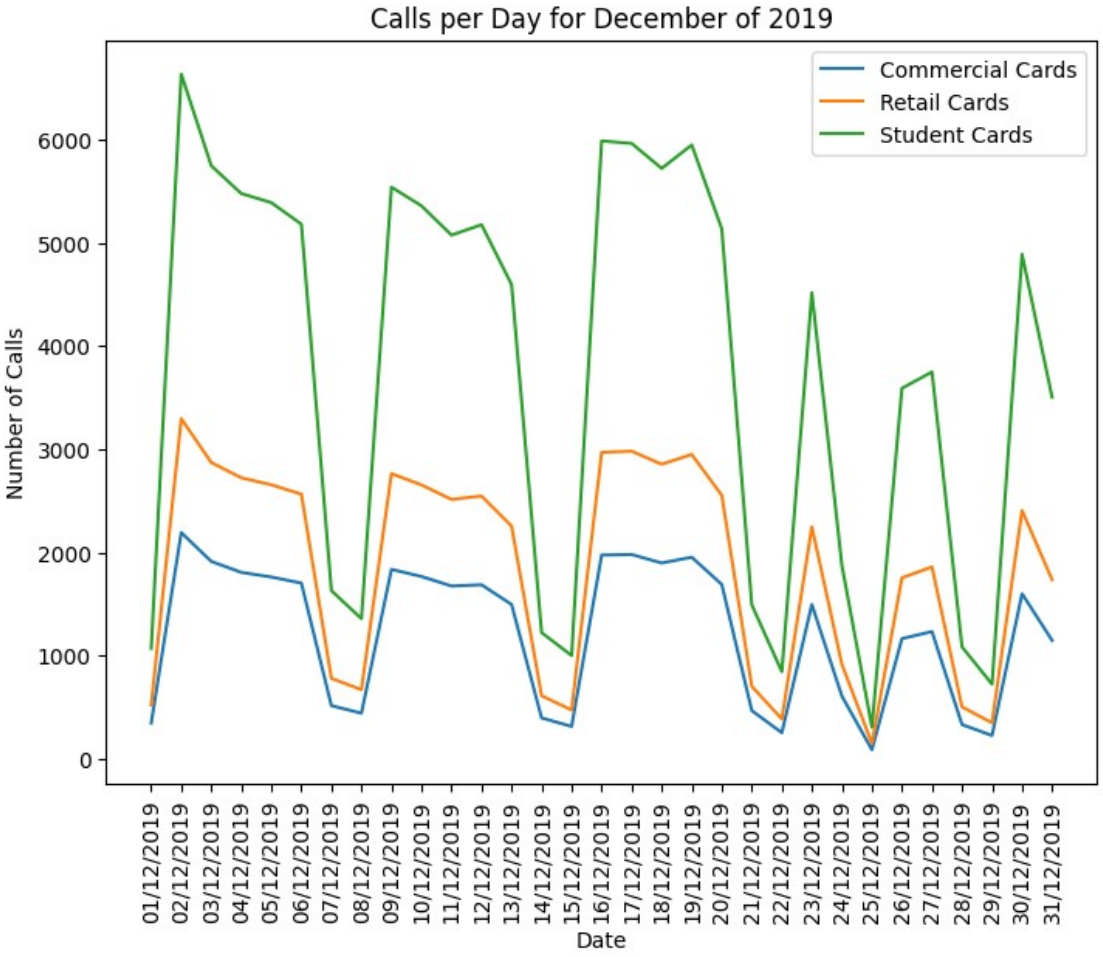

Calls per Day for July of 2019
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/07/2019
02/07/2019
03/07/2019
04/07/2019
05/07/2019
06/07/2019
07/07/2019
08/07/2019
09/07/2019
10/07/2019
11/07/2019
12/07/2019
13/07/2019
14/07/2019
15/07/2019
16/07/2019
17/07/2019
18/07/2019
19/07/2019
20/07/2019
21/07/2019
22/07/2019
23/07/2019
24/07/2019
25/07/2019
26/07/2019
27/07/2019
28/07/2019
29/07/2019
30/07/2019
31/07/2019
Date

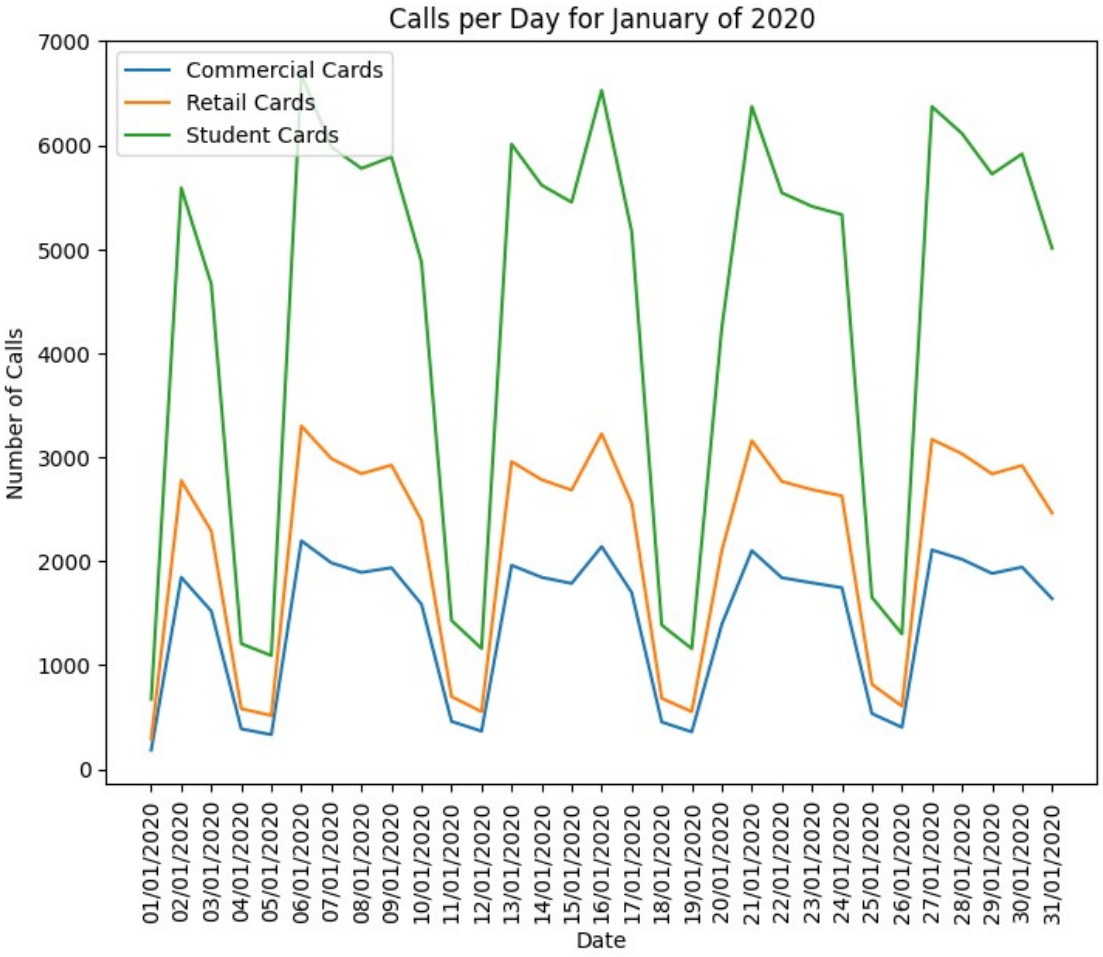

Calls per Day for August of 2019
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
Commercial Cards
Retail Cards
Student Cards
01/08/2019
02/08/2019
03/08/2019
04/08/2019
05/08/2019
06/08/2019
07/08/2019
08/08/2019
09/08/2019
10/08/2019
11/08/2019
12/08/2019
13/08/2019
14/08/2019
15/08/2019
16/08/2019
17/08/2019
18/08/2019
19/08/2019
20/08/2019
21/08/2019
22/08/2019
23/08/2019
24/08/2019
25/08/2019
26/08/2019
27/08/2019
28/08/2019
29/08/2019
30/08/2019
31/08/2019
Date

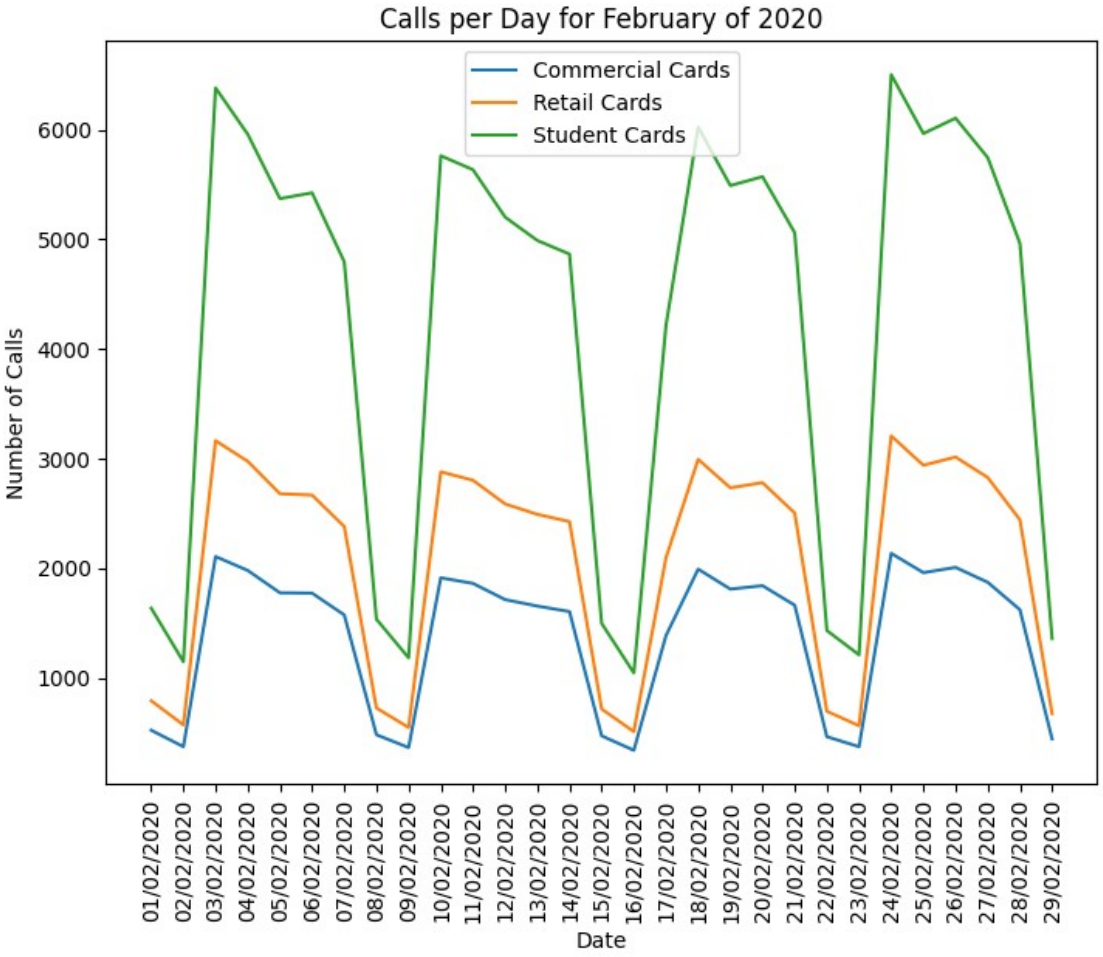

Calls per Day for September of 2019
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/09/2019
02/09/2019
03/09/2019
04/09/2019
05/09/2019
06/09/2019
07/09/2019
08/09/2019
09/09/2019
10/09/2019
11/09/2019
12/09/2019
13/09/2019
14/09/2019
15/09/2019
16/09/2019
17/09/2019
18/09/2019
19/09/2019
20/09/2019
21/09/2019
22/09/2019
23/09/2019
24/09/2019
25/09/2019
26/09/2019
27/09/2019
28/09/2019
29/09/2019
30/09/2019
Date

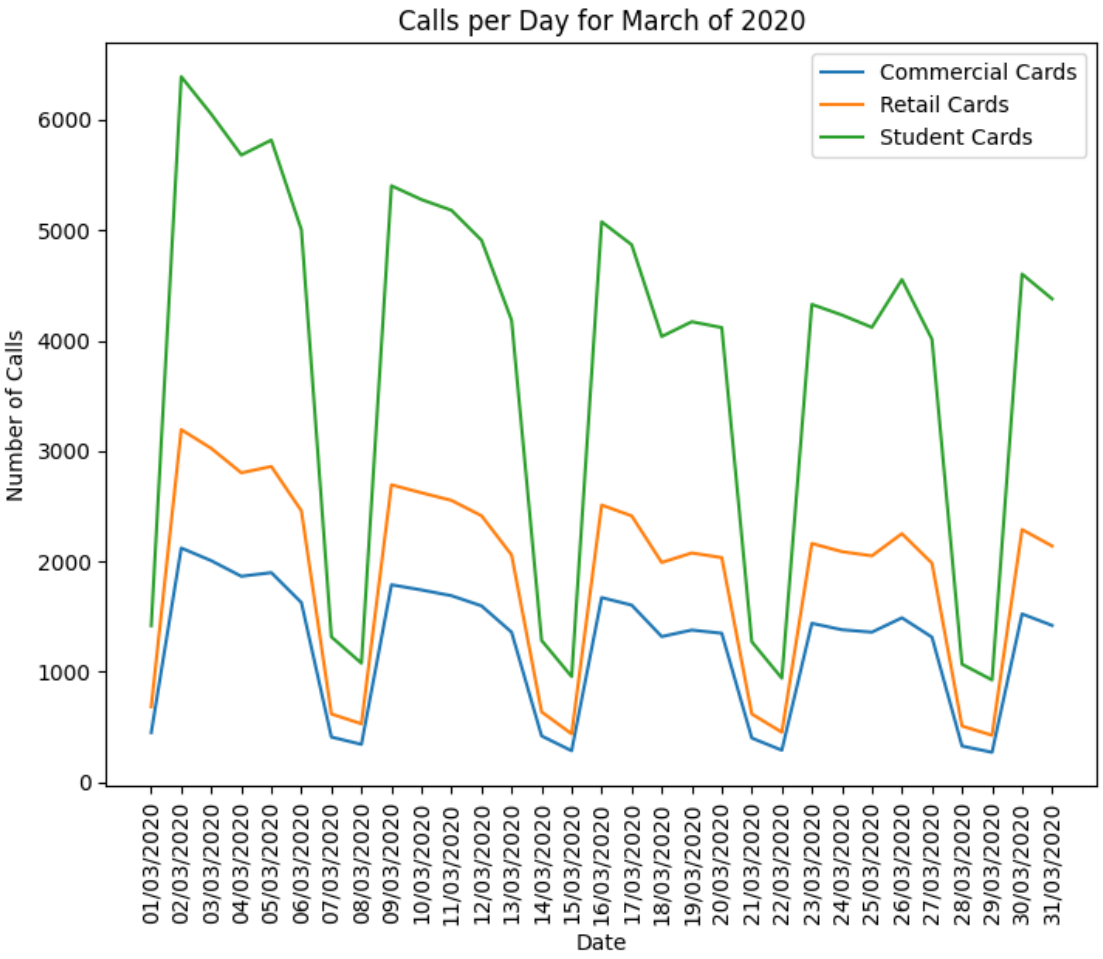

Calls per Day for October of 2019
Commercial Cards
Retail Cards
Student Cards
Number of Calls
7000
6000
5000
4000
3000
2000
1000
01/10/2019
02/10/2019
03/10/2019
04/10/2019
05/10/2019
06/10/2019
07/10/2019
08/10/2019
09/10/2019
10/10/2019
11/10/2019
12/10/2019
13/10/2019
14/10/2019
15/10/2019
16/10/2019
17/10/2019
18/10/2019
19/10/2019
20/10/2019
21/10/2019
22/10/2019
23/10/2019
24/10/2019
25/10/2019
26/10/2019
27/10/2019
28/10/2019
29/10/2019
30/10/2019
31/10/2019
Date

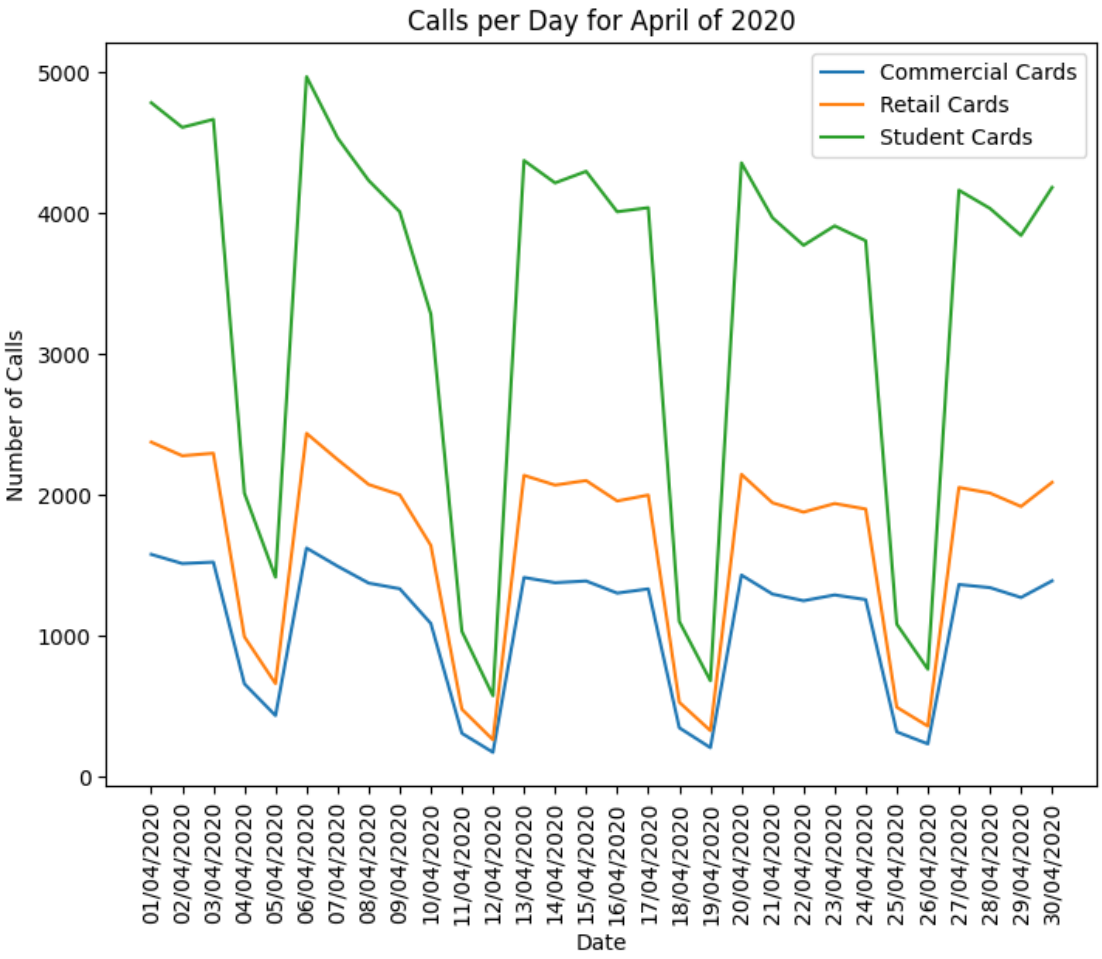

Calls per Day for November of 2019
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/11/2019
02/11/2019
03/11/2019
04/11/2019
05/11/2019
06/11/2019
07/11/2019
08/11/2019
09/11/2019
10/11/2019
11/11/2019
12/11/2019
13/11/2019
14/11/2019
15/11/2019
16/11/2019
17/11/2019
18/11/2019
19/11/2019
20/11/2019
21/11/2019
22/11/2019
23/11/2019
24/11/2019
25/11/2019
26/11/2019
27/11/2019
28/11/2019
29/11/2019
30/11/2019
Date

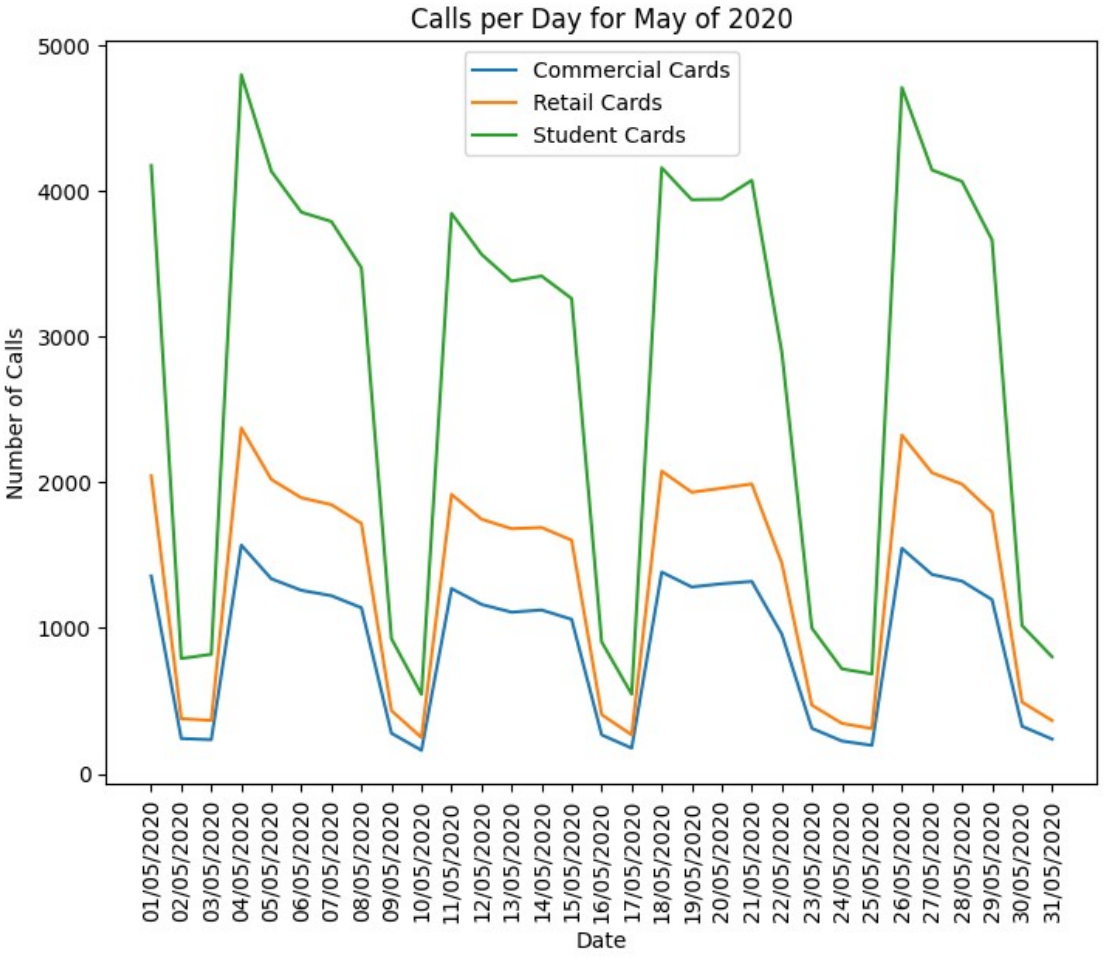

Calls per Day for December of 2019
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
01/12/2019
02/12/2019
03/12/2019
04/12/2019
05/12/2019
06/12/2019
07/12/2019
08/12/2019
09/12/2019
10/12/2019
11/12/2019
12/12/2019
13/12/2019
14/12/2019
15/12/2019
16/12/2019
17/12/2019
18/12/2019
19/12/2019
20/12/2019
21/12/2019
22/12/2019
23/12/2019
24/12/2019
25/12/2019
26/12/2019
27/12/2019
28/12/2019
29/12/2019
30/12/2019
31/12/2019
Date

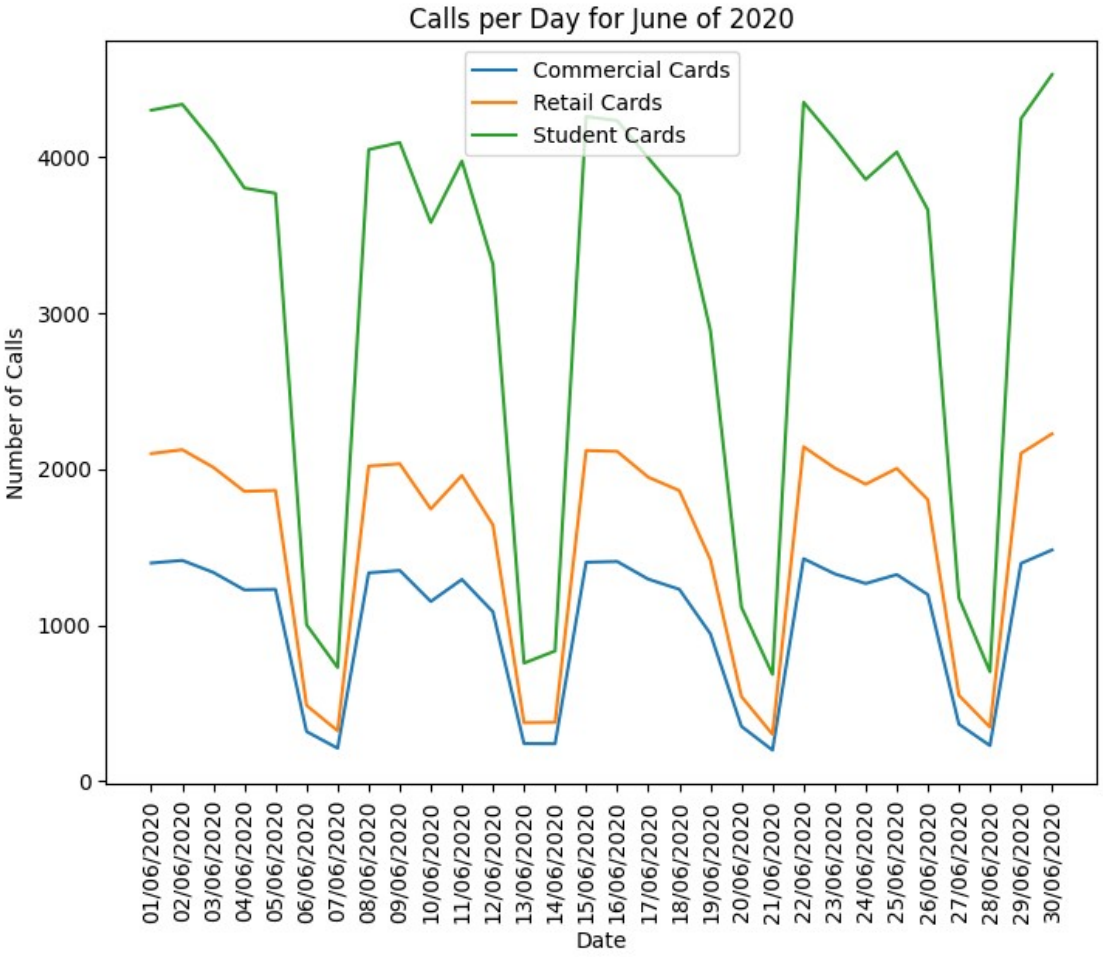

Calls per Day for January of 2020
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/01/2020
02/01/2020
03/01/2020
04/01/2020
05/01/2020
06/01/2020
07/01/2020
08/01/2020
09/01/2020
10/01/2020
11/01/2020
12/01/2020
13/01/2020
14/01/2020
15/01/2020
16/01/2020
17/01/2020
18/01/2020
19/01/2020
20/01/2020
21/01/2020
22/01/2020
23/01/2020
24/01/2020
25/01/2020
26/01/2020
27/01/2020
28/01/2020
29/01/2020
30/01/2020
31/01/2020
Date

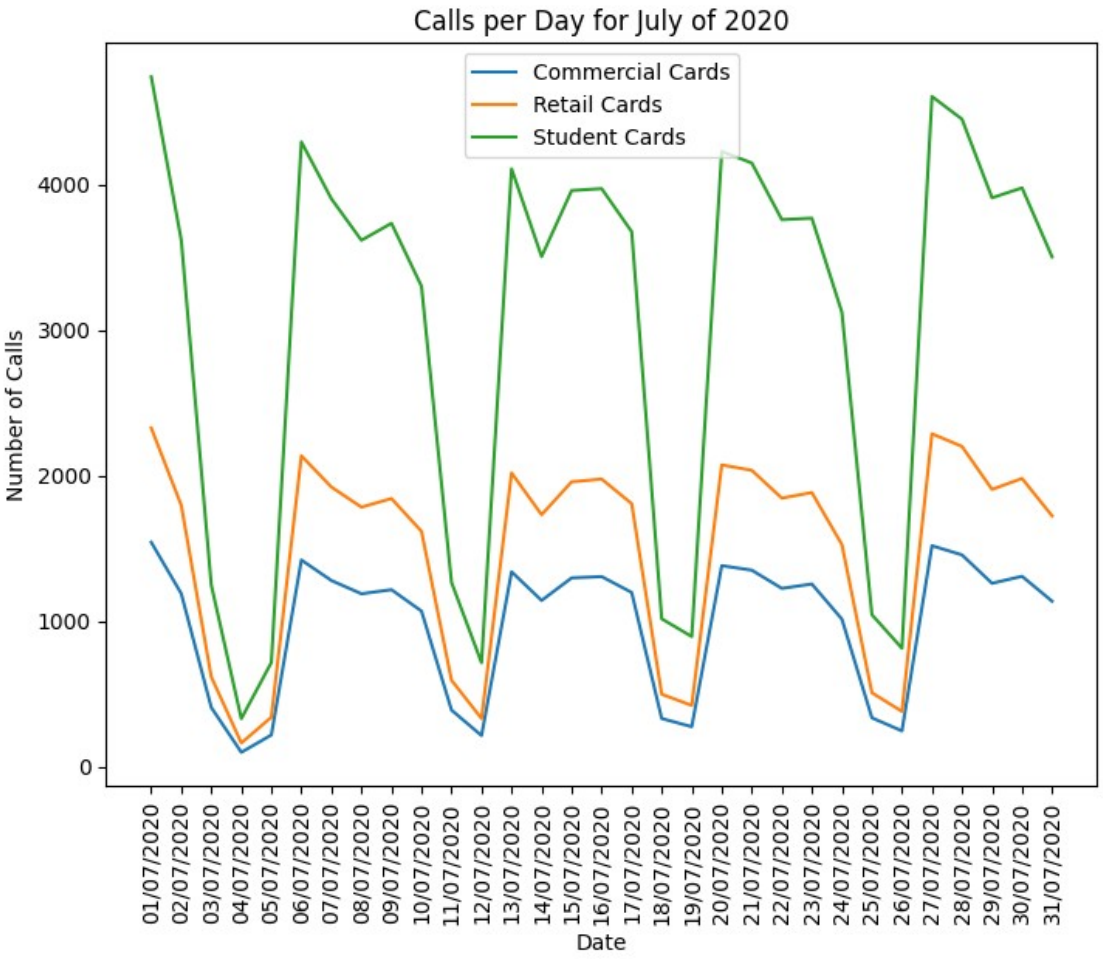

Calls per Day for February of 2020
Commercial Cards
Retail Cards
Student Cards
Number of Calls
1000
2000
3000
4000
5000
6000
01/02/2020
02/02/2020
03/02/2020
04/02/2020
05/02/2020
06/02/2020
07/02/2020
08/02/2020
09/02/2020
10/02/2020
11/02/2020
12/02/2020
13/02/2020
14/02/2020
15/02/2020
16/02/2020
17/02/2020
18/02/2020
19/02/2020
20/02/2020
21/02/2020
22/02/2020
23/02/2020
24/02/2020
25/02/2020
26/02/2020
27/02/2020
28/02/2020
29/02/2020
Date

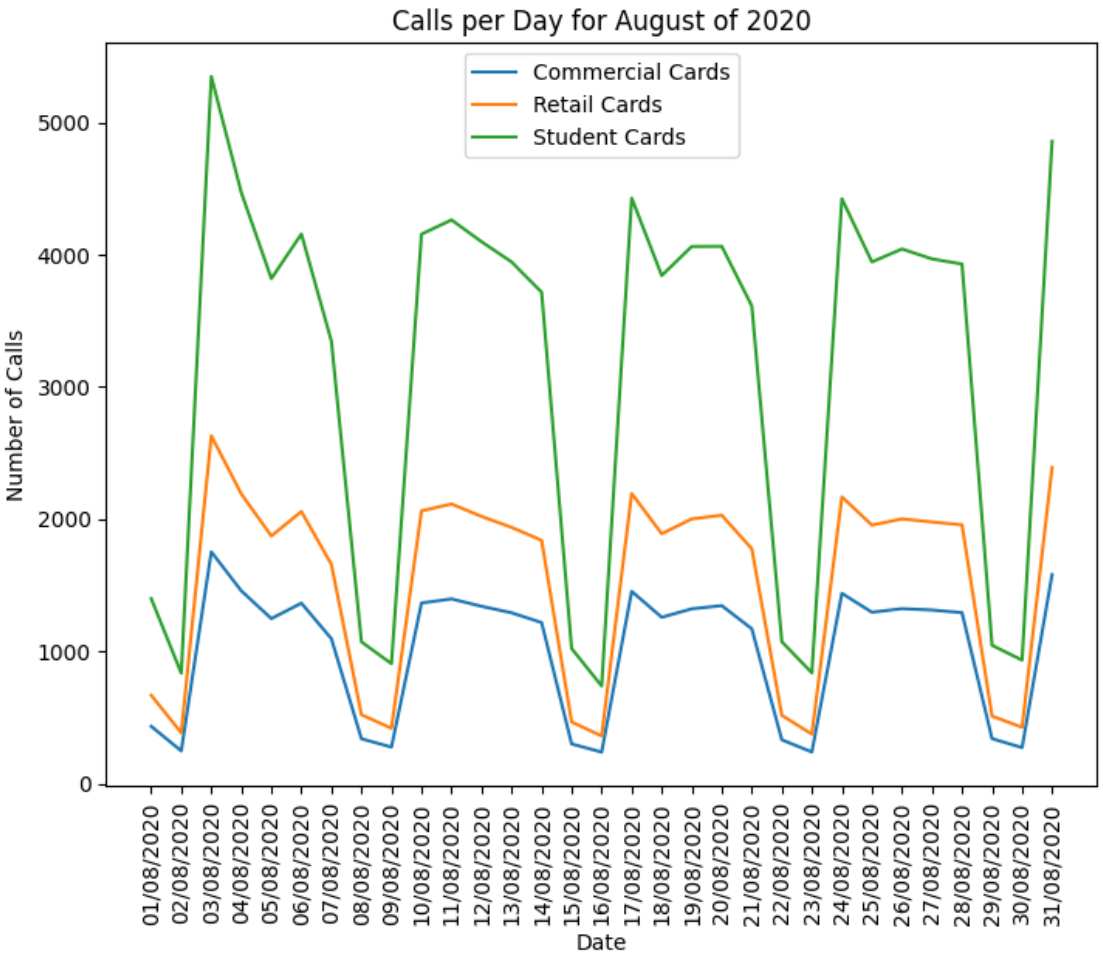

Calls per Day for March of 2020
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
6000
01/03/2020
02/03/2020
03/03/2020
04/03/2020
05/03/2020
06/03/2020
07/03/2020
08/03/2020
09/03/2020
10/03/2020
11/03/2020
12/03/2020
13/03/2020
14/03/2020
15/03/2020
16/03/2020
17/03/2020
18/03/2020
19/03/2020
20/03/2020
21/03/2020
22/03/2020
23/03/2020
24/03/2020
25/03/2020
26/03/2020
27/03/2020
28/03/2020
29/03/2020
30/03/2020
31/03/2020
Date

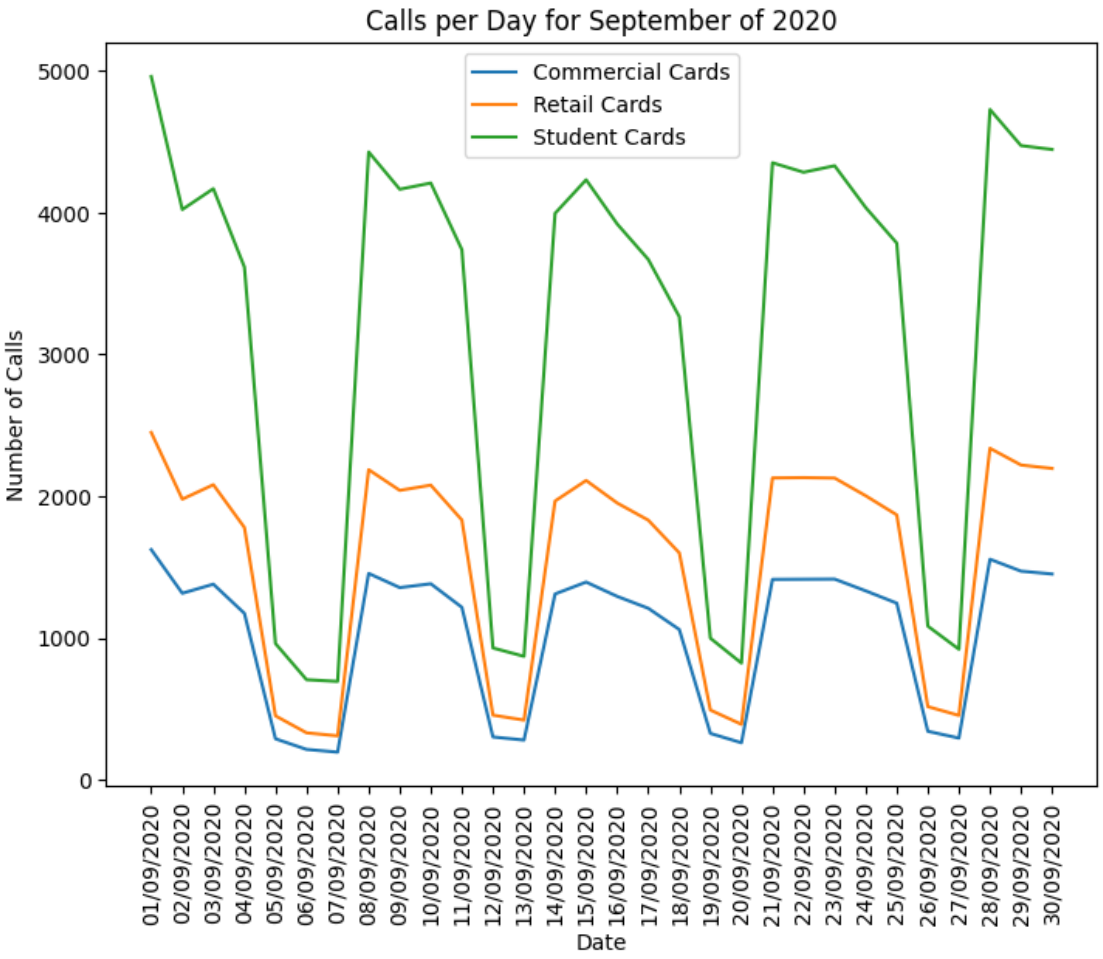

Calls per Day for April of 2020
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
01/04/2020
02/04/2020
03/04/2020
04/04/2020
05/04/2020
06/04/2020
07/04/2020
08/04/2020
09/04/2020
10/04/2020
11/04/2020
12/04/2020
13/04/2020
14/04/2020
15/04/2020
16/04/2020
17/04/2020
18/04/2020
19/04/2020
20/04/2020
21/04/2020
22/04/2020
23/04/2020
24/04/2020
25/04/2020
26/04/2020
27/04/2020
28/04/2020
29/04/2020
30/04/2020
Date

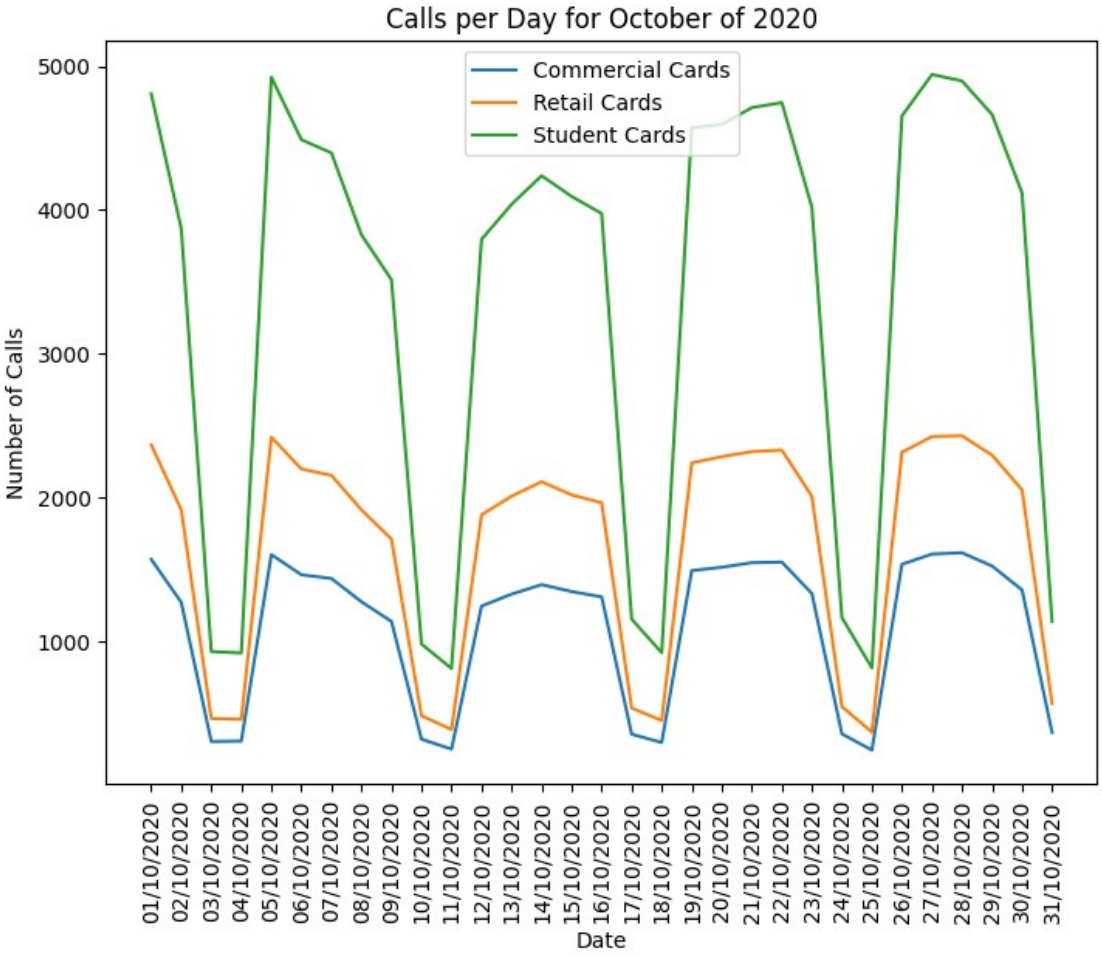

Calls per Day for May of 2020
Commercial Cards
Retail Cards
Student Cards
Number of Calls
5000
4000
3000
2000
1000
0
01/05/2020
02/05/2020
03/05/2020
04/05/2020
05/05/2020
06/05/2020
07/05/2020
08/05/2020
09/05/2020
10/05/2020
11/05/2020
12/05/2020
13/05/2020
14/05/2020
15/05/2020
16/05/2020
17/05/2020
18/05/2020
19/05/2020
20/05/2020
21/05/2020
22/05/2020
23/05/2020
24/05/2020
25/05/2020
26/05/2020
27/05/2020
28/05/2020
29/05/2020
30/05/2020
31/05/2020
Date

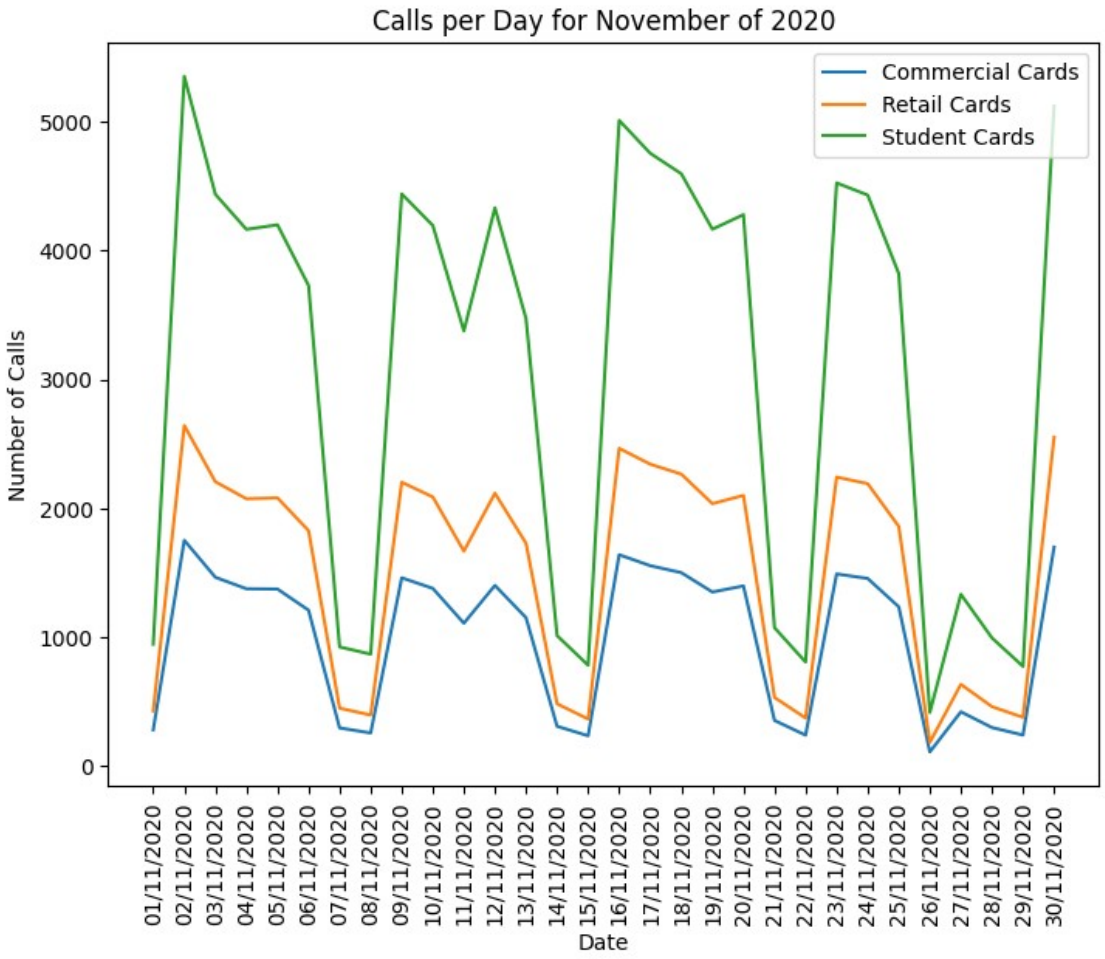

Calls per Day for June of 2020
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
01/06/2020
02/06/2020
03/06/2020
04/06/2020
05/06/2020
06/06/2020
07/06/2020
08/06/2020
09/06/2020
10/06/2020
11/06/2020
12/06/2020
13/06/2020
14/06/2020
15/06/2020
16/06/2020
17/06/2020
18/06/2020
19/06/2020
20/06/2020
21/06/2020
22/06/2020
23/06/2020
24/06/2020
25/06/2020
26/06/2020
27/06/2020
28/06/2020
29/06/2020
30/06/2020
Date

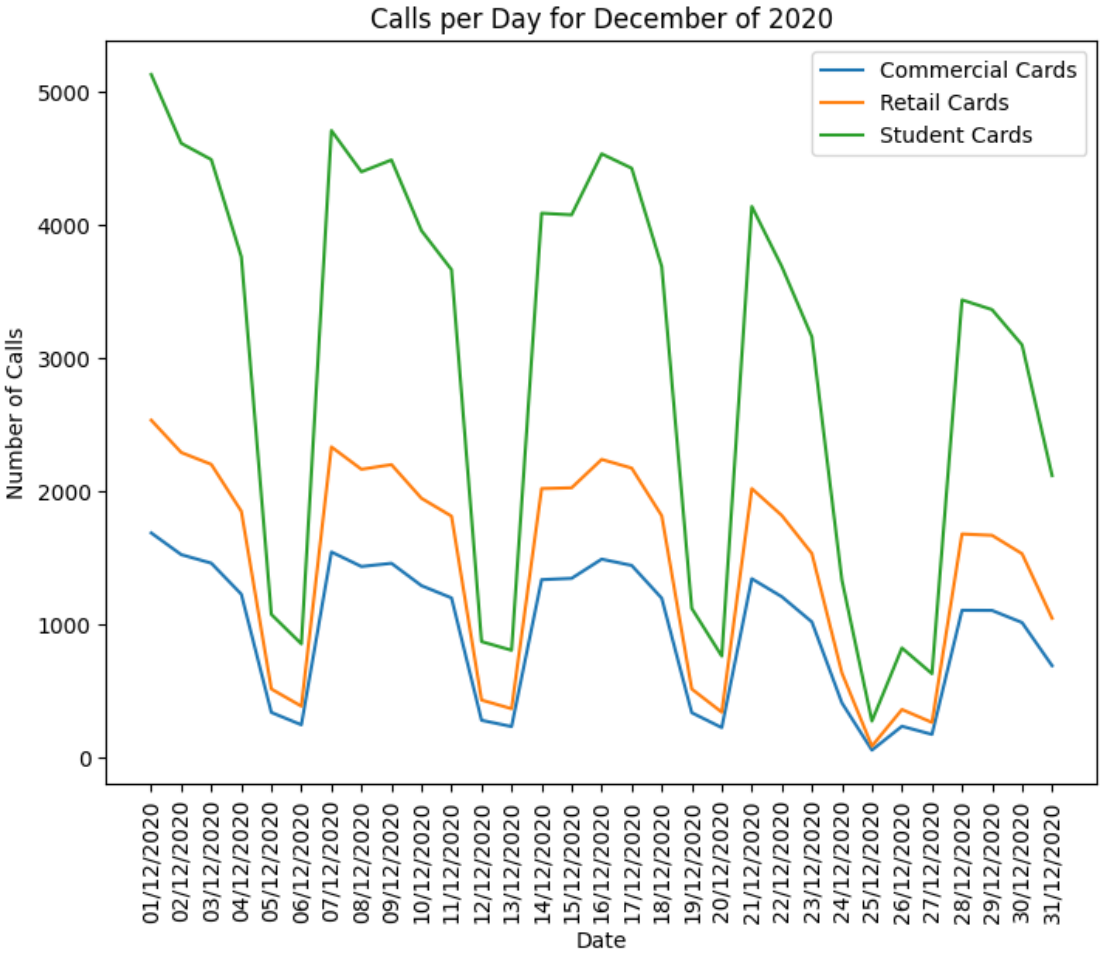

Calls per Day for July of 2020
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
01/07/2020
02/07/2020
03/07/2020
04/07/2020
05/07/2020
06/07/2020
07/07/2020
08/07/2020
09/07/2020
10/07/2020
11/07/2020
12/07/2020
13/07/2020
14/07/2020
15/07/2020
16/07/2020
17/07/2020
18/07/2020
19/07/2020
20/07/2020
21/07/2020
22/07/2020
23/07/2020
24/07/2020
25/07/2020
26/07/2020
27/07/2020
28/07/2020
29/07/2020
30/07/2020
31/07/2020
Date

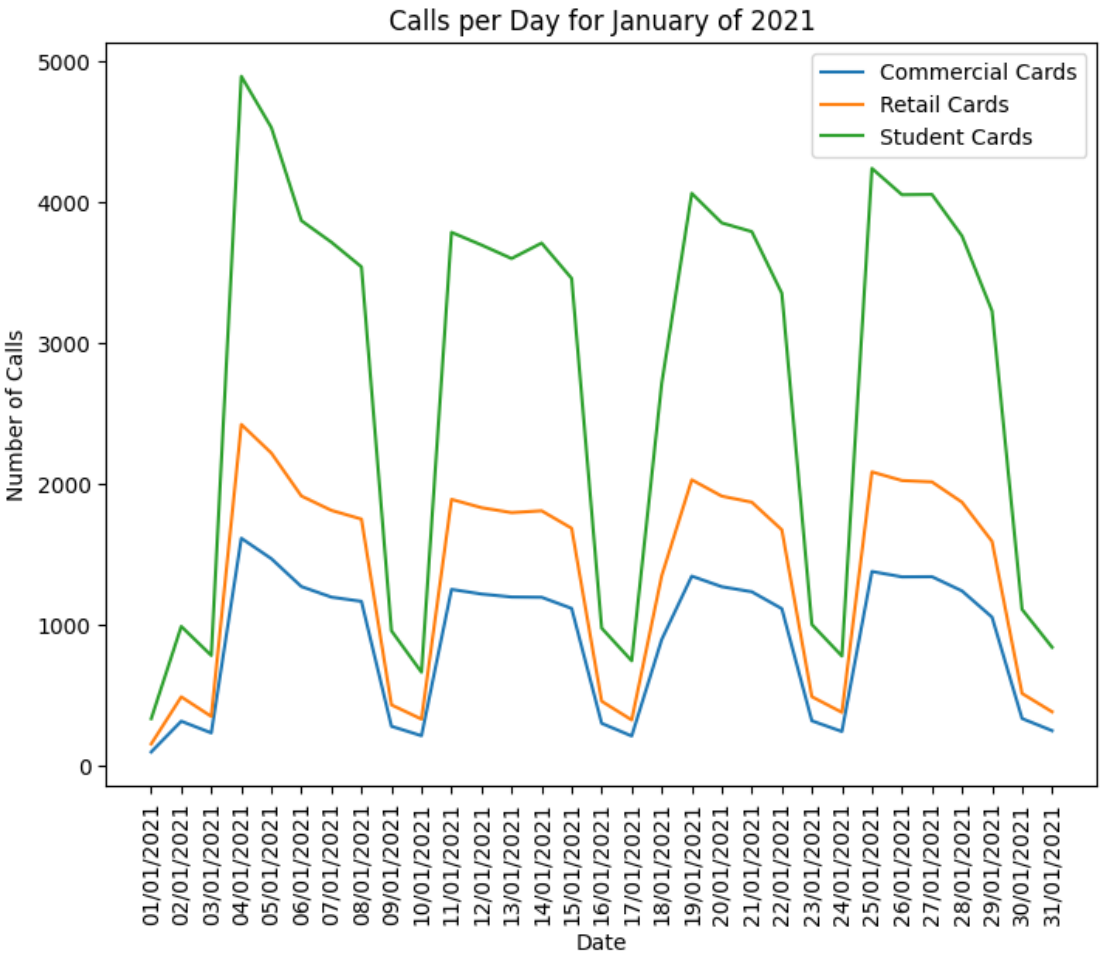

Calls per Day for August of 2020
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
01/08/2020
02/08/2020
03/08/2020
04/08/2020
05/08/2020
06/08/2020
07/08/2020
08/08/2020
09/08/2020
10/08/2020
11/08/2020
12/08/2020
13/08/2020
14/08/2020
15/08/2020
16/08/2020
17/08/2020
18/08/2020
19/08/2020
20/08/2020
21/08/2020
22/08/2020
23/08/2020
24/08/2020
25/08/2020
26/08/2020
27/08/2020
28/08/2020
29/08/2020
30/08/2020
31/08/2020
Date

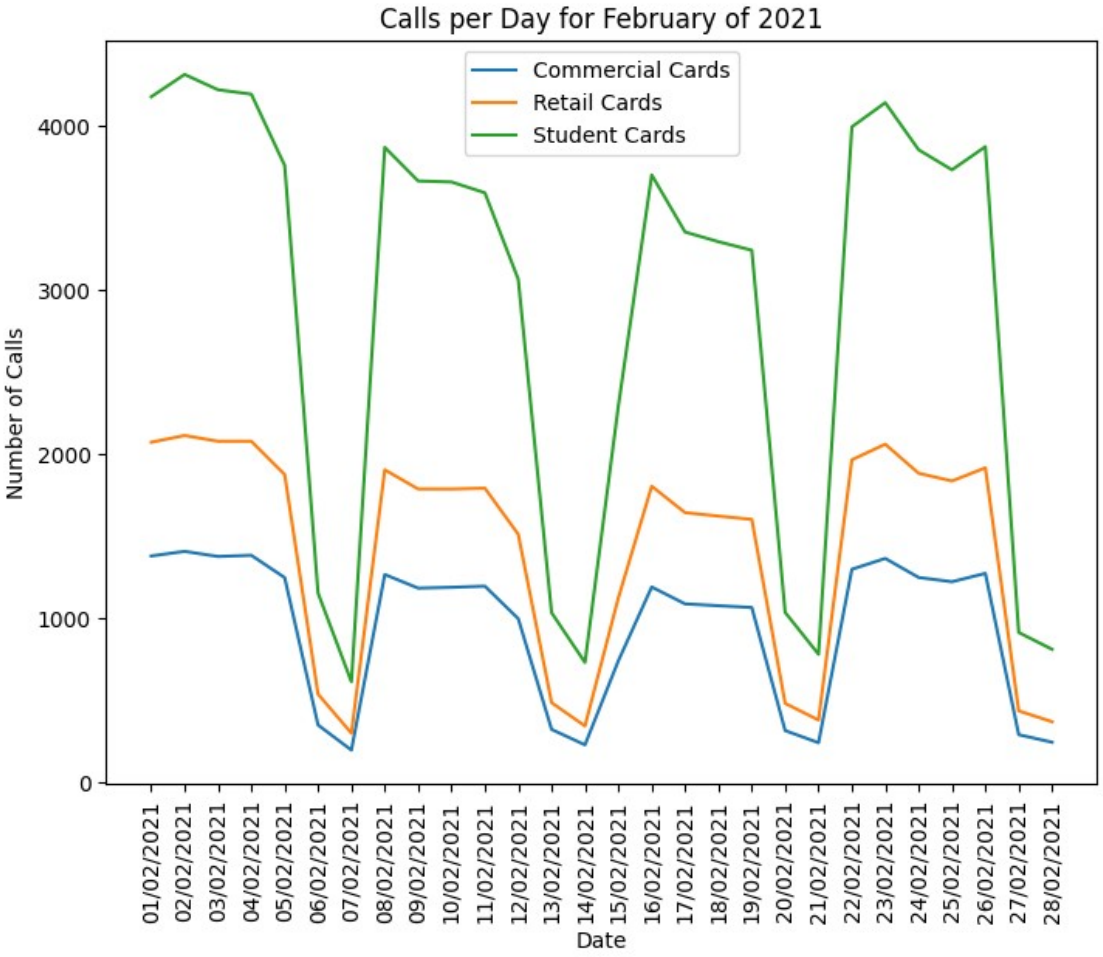

Calls per Day for September of 2020
Commercial Cards
Retail Cards
Student Cards
Number of Calls
5000
4000
3000
2000
1000
0
01/09/2020
02/09/2020
03/09/2020
04/09/2020
05/09/2020
06/09/2020
07/09/2020
08/09/2020
09/09/2020
10/09/2020
11/09/2020
12/09/2020
13/09/2020
14/09/2020
15/09/2020
16/09/2020
17/09/2020
18/09/2020
19/09/2020
20/09/2020
21/09/2020
22/09/2020
23/09/2020
24/09/2020
25/09/2020
26/09/2020
27/09/2020
28/09/2020
29/09/2020
30/09/2020
Date

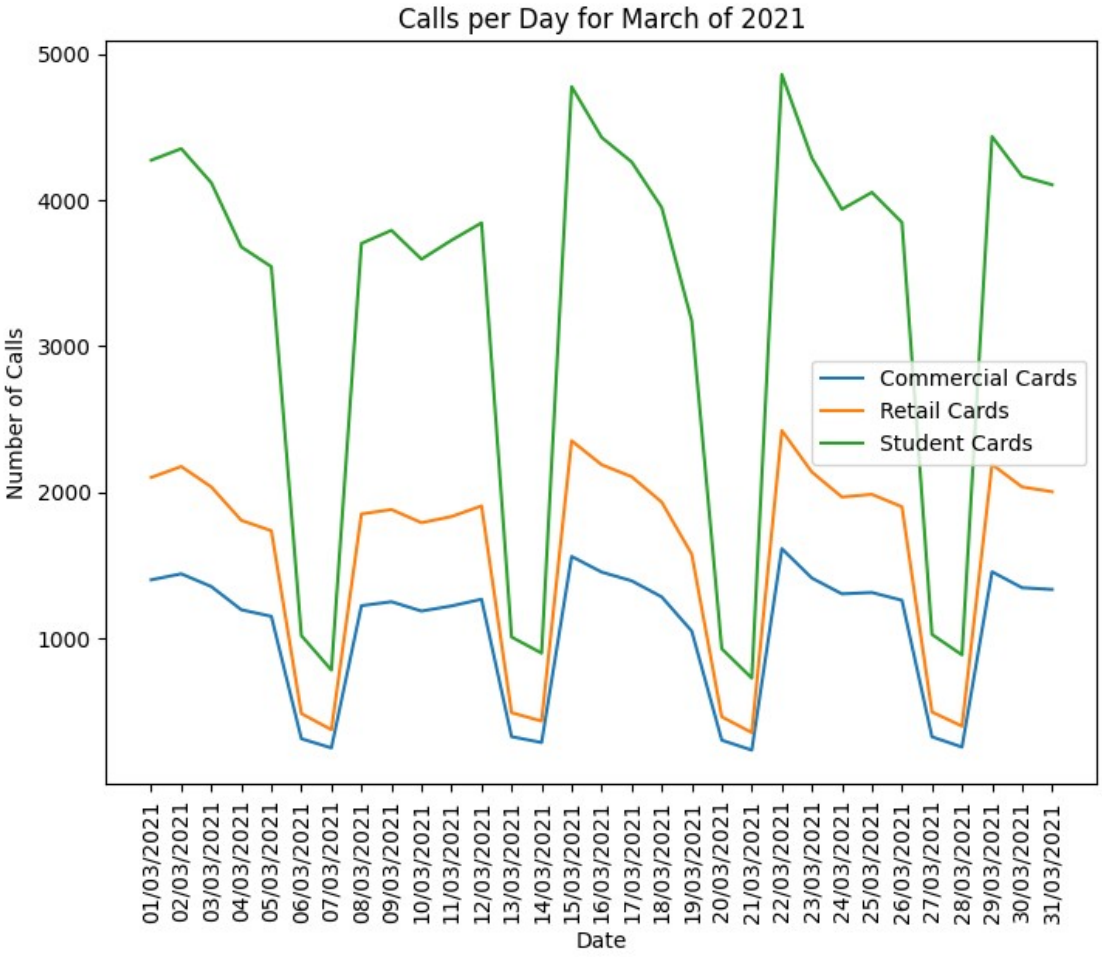

Calls per Day for October of 2020
Commercial Cards
Retail Cards
Student Cards
Number of Calls
5000
4000
3000
2000
1000
01/10/2020
02/10/2020
03/10/2020
04/10/2020
05/10/2020
06/10/2020
07/10/2020
08/10/2020
09/10/2020
10/10/2020
11/10/2020
12/10/2020
13/10/2020
14/10/2020
15/10/2020
16/10/2020
17/10/2020
18/10/2020
19/10/2020
20/10/2020
21/10/2020
22/10/2020
23/10/2020
24/10/2020
25/10/2020
26/10/2020
27/10/2020
28/10/2020
29/10/2020
30/10/2020
31/10/2020
Date

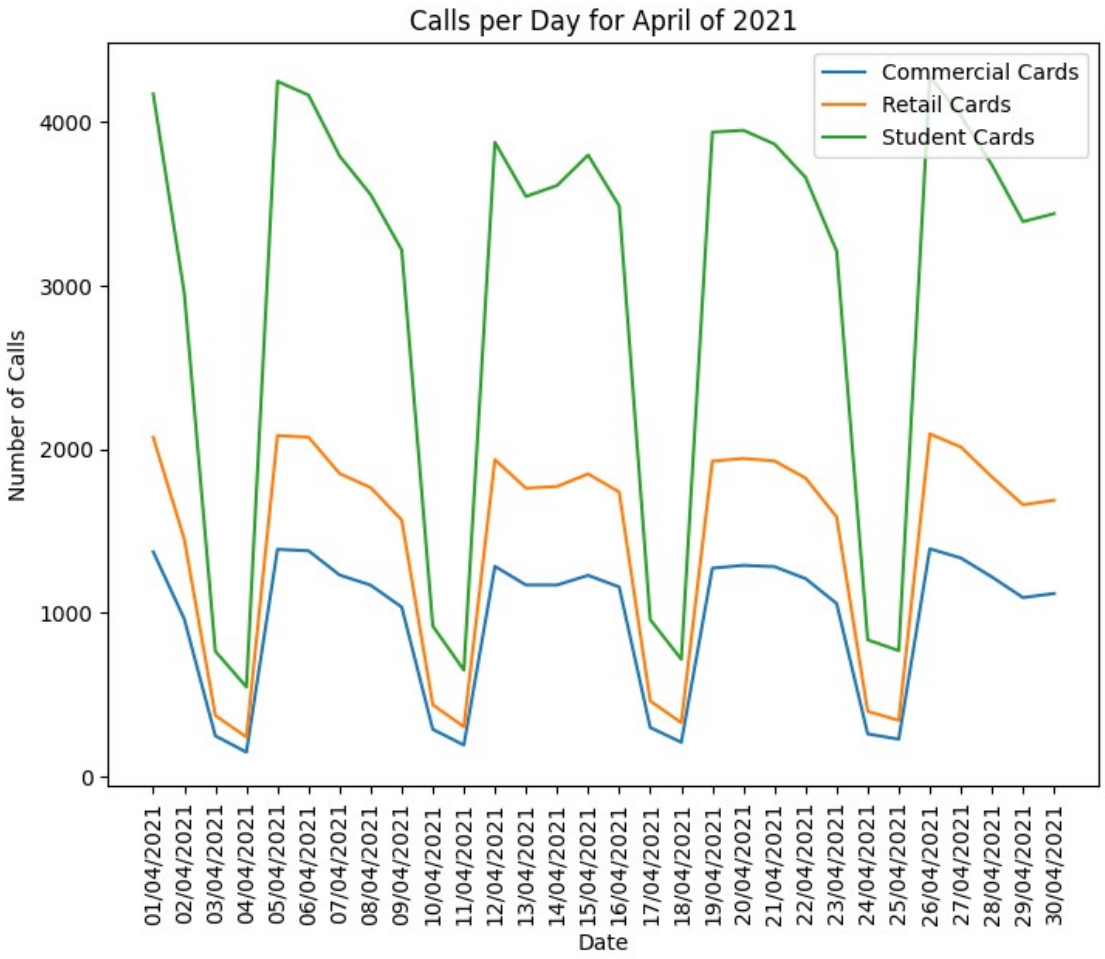

Calls per Day for November of 2020
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
01/11/2020
02/11/2020
03/11/2020
04/11/2020
05/11/2020
06/11/2020
07/11/2020
08/11/2020
09/11/2020
10/11/2020
11/11/2020
12/11/2020
13/11/2020
14/11/2020
15/11/2020
16/11/2020
17/11/2020
18/11/2020
19/11/2020
20/11/2020
21/11/2020
22/11/2020
23/11/2020
24/11/2020
25/11/2020
26/11/2020
27/11/2020
28/11/2020
29/11/2020
30/11/2020
Date


Calls per Day for December of 2020
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
5000
01/12/2020
02/12/2020
03/12/2020
04/12/2020
05/12/2020
06/12/2020
07/12/2020
08/12/2020
09/12/2020
10/12/2020
11/12/2020
12/12/2020
13/12/2020
14/12/2020
15/12/2020
16/12/2020
17/12/2020
18/12/2020
19/12/2020
20/12/2020
21/12/2020
22/12/2020
23/12/2020
24/12/2020
25/12/2020
26/12/2020
27/12/2020
28/12/2020
29/12/2020
30/12/2020
31/12/2020
Date

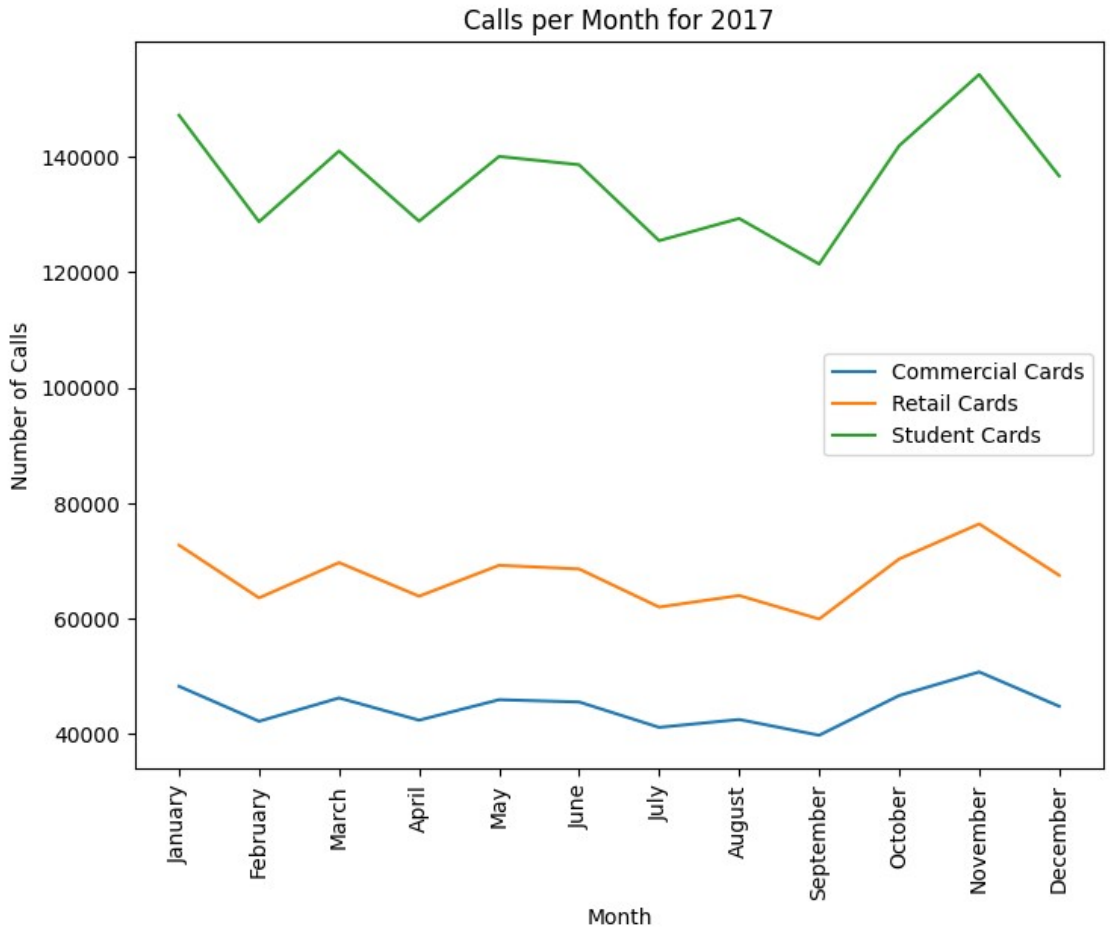

Calls per Day for January of 2021
Commercial Cards
Retail Cards
Student Cards
Number of Calls
5000
4000
3000
2000
1000
0
01/01/2021
02/01/2021
03/01/2021
04/01/2021
05/01/2021
06/01/2021
07/01/2021
08/01/2021
09/01/2021
10/01/2021
11/01/2021
12/01/2021
13/01/2021
14/01/2021
15/01/2021
16/01/2021
17/01/2021
18/01/2021
19/01/2021
20/01/2021
21/01/2021
22/01/2021
23/01/2021
24/01/2021
25/01/2021
26/01/2021
27/01/2021
28/01/2021
29/01/2021
30/01/2021
31/01/2021
Date


Calls per Day for February of 2021
Commercial Cards
Retail Cards
Student Cards
Number of Calls
0
1000
2000
3000
4000
01/02/2021
02/02/2021
03/02/2021
04/02/2021
05/02/2021
06/02/2021
07/02/2021
08/02/2021
09/02/2021
10/02/2021
11/02/2021
12/02/2021
13/02/2021
14/02/2021
15/02/2021
16/02/2021
17/02/2021
18/02/2021
19/02/2021
20/02/2021
21/02/2021
22/02/2021
23/02/2021
24/02/2021
25/02/2021
26/02/2021
27/02/2021
28/02/2021
Date


Calls per Day for March of 2021
Number of Calls
5000
4000
3000
2000
1000
Commercial Cards
Retail Cards
Student Cards
01/03/2021
02/03/2021
03/03/2021
04/03/2021
05/03/2021
06/03/2021
07/03/2021
08/03/2021
09/03/2021
10/03/2021
11/03/2021
12/03/2021
13/03/2021
14/03/2021
15/03/2021
16/03/2021
17/03/2021
18/03/2021
19/03/2021
20/03/2021
21/03/2021
22/03/2021
23/03/2021
24/03/2021
25/03/2021
26/03/2021
27/03/2021
28/03/2021
29/03/2021
30/03/2021
31/03/2021
Date

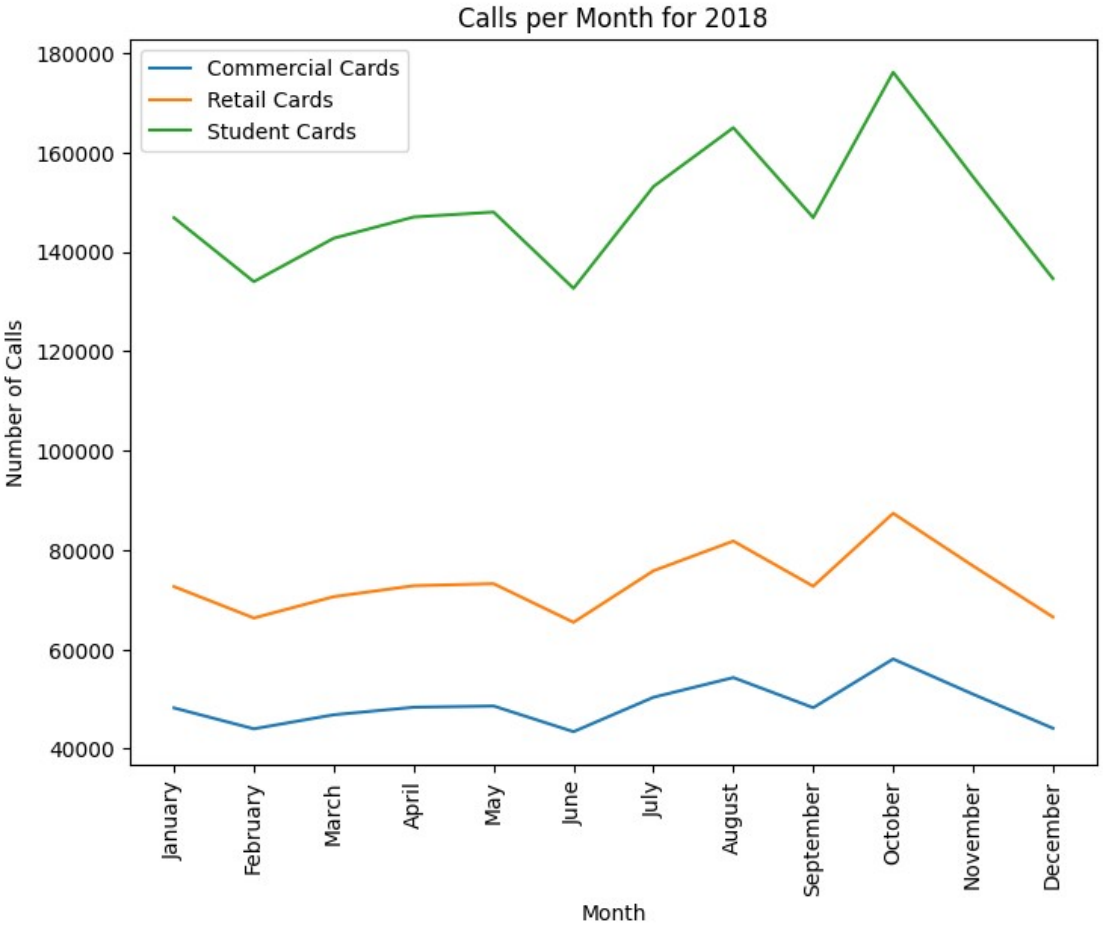


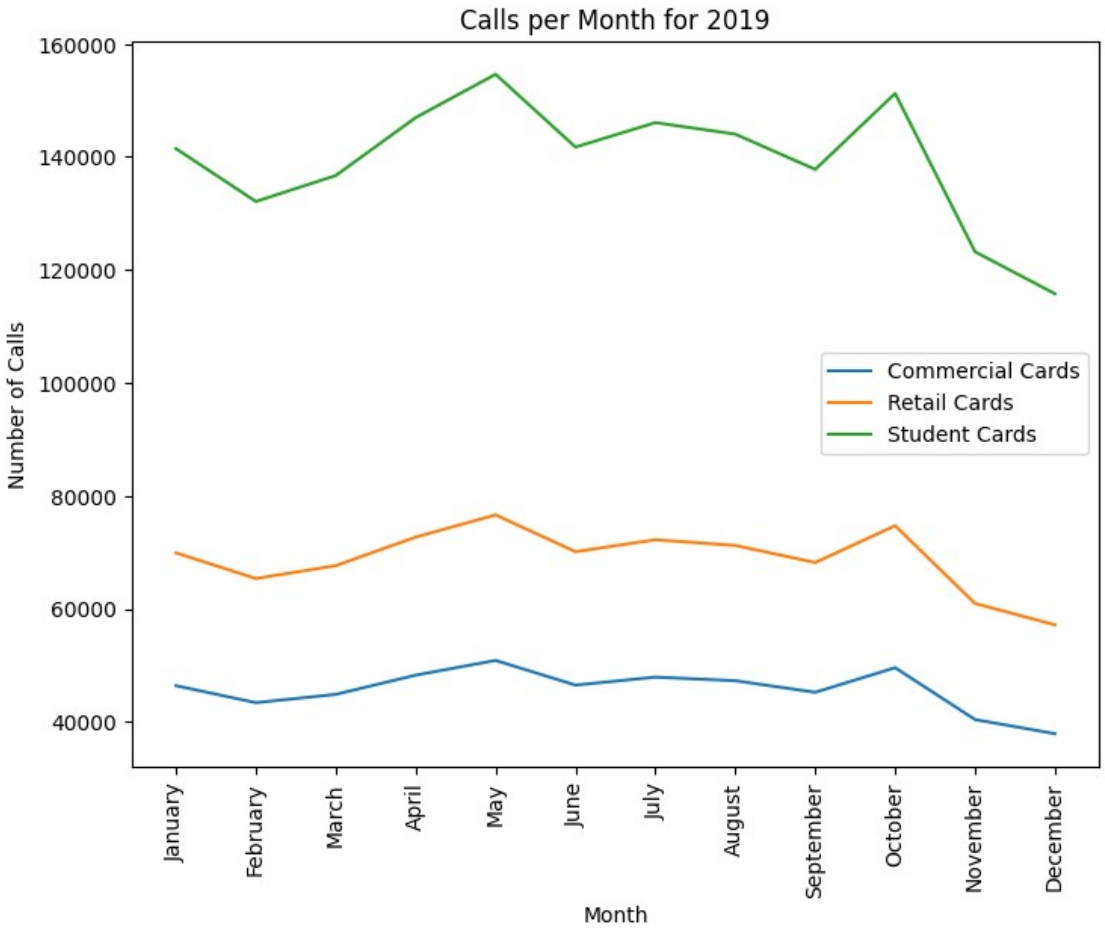

```
# Cumulative Montly Data
df_17_cumm = df_17[['cc', 'rc' , 'sc',
'month']].groupby('month').sum()

# Plot Cumulative Monthly Data for 2017
x = range(len(df_17_cumm))
cc_values = df_17_cumm.cc
rc_values = df_17_cumm.rc
sc_values = df_17_cumm.sc
fig = plt.figure()
ax = fig.add_axes([1,1,1,1])
ax.plot(x, cc_values, label='Commercial Cards')
ax.plot(x, rc_values, label='Retail Cards')
ax.plot(x, sc_values, label='Student Cards')
ax.set_xticks(x, [ arrow.get(str(m), "M").format("MMMM") for m in
df_17_cumm.index ], rotation=90)
ax.set_xlabel("Month")
ax.set_ylabel("Number of Calls")
ax.set_title(f"Calls per Month for 2017")
ax.legend()
plt.show()
```

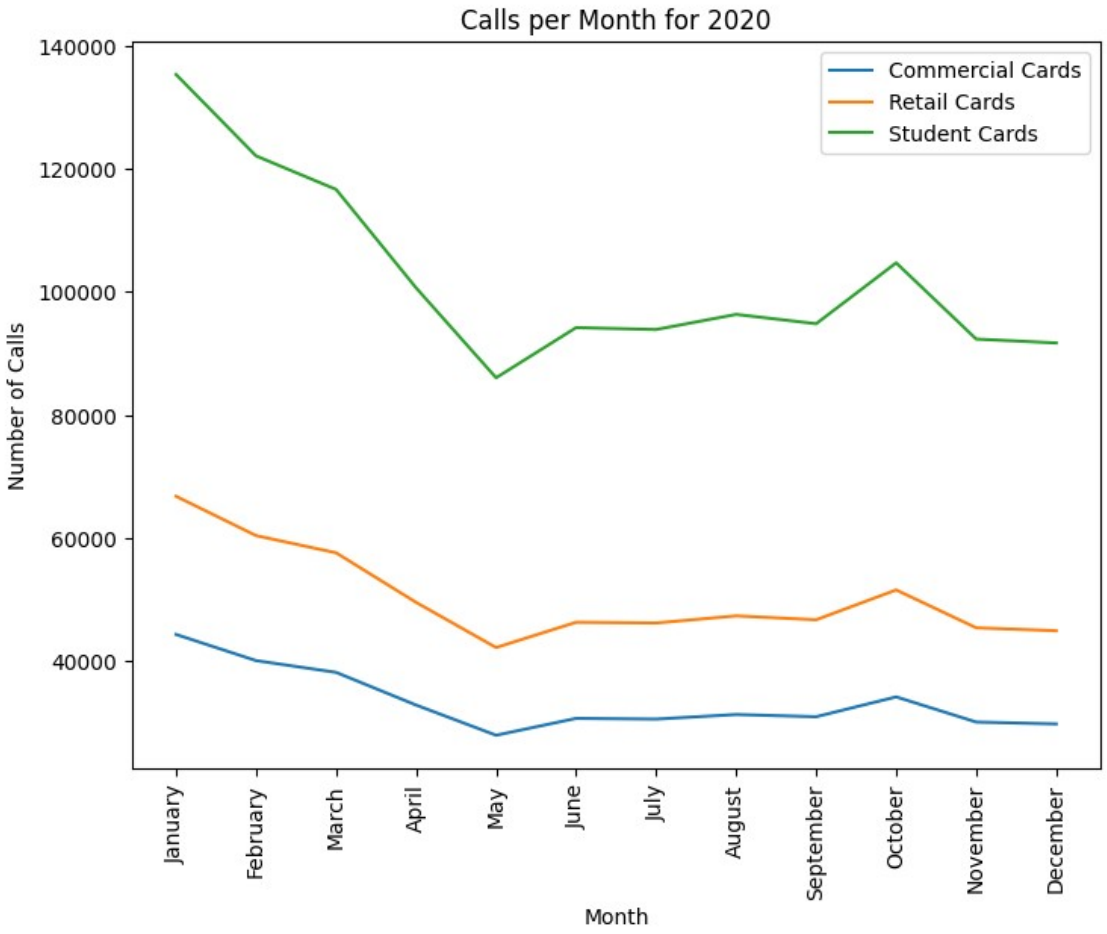


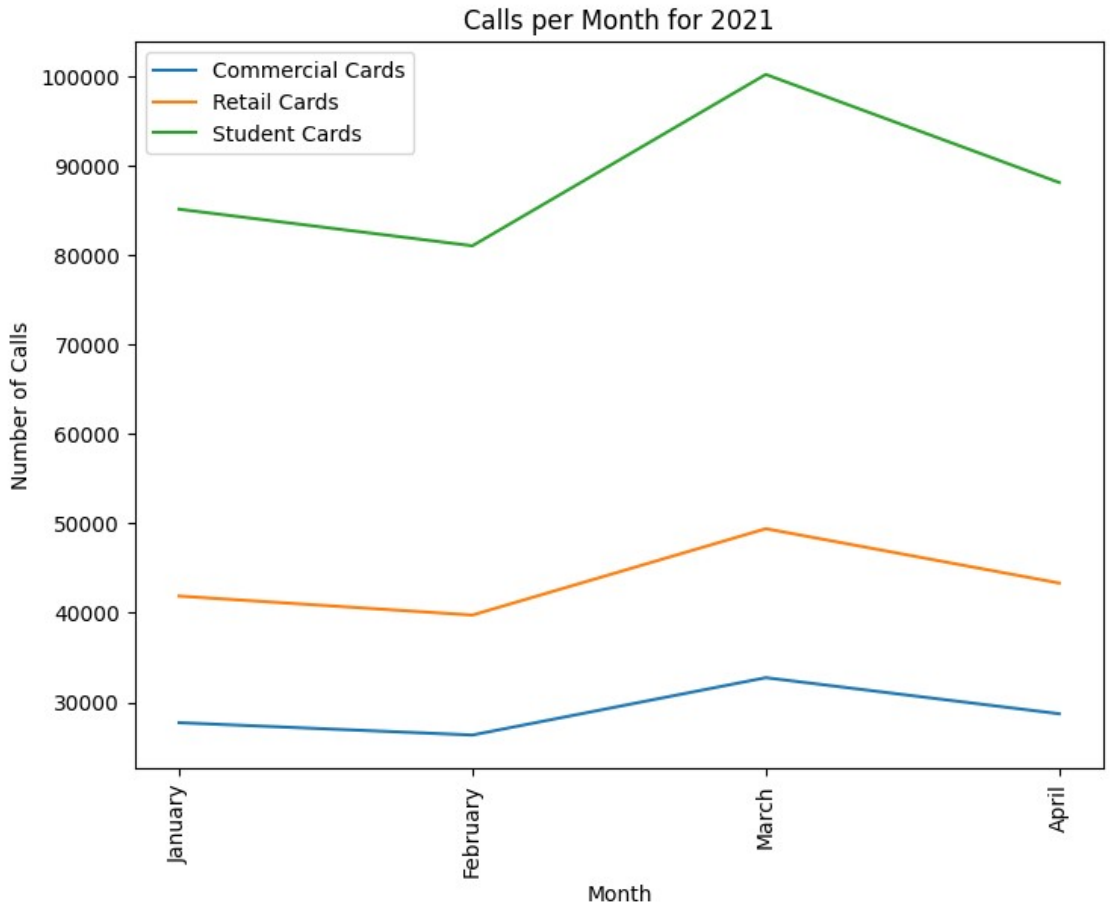

```
# Plot Cumulative Monthly Data for 2018-2021
years = df_not17_sorted.year.unique()
for y in years:
    df_not17_sorted_yearly = df_not17_sorted[df_not17_sorted.year ==
y]
    df_cumm = df_not17_sorted_yearly[['cc', 'rc' , 'sc',
'month']].groupby('month').sum()
    x = range(len(df_cumm))
    cc_values = df_cumm.cc
    rc_values = df_cumm.rc
    sc_values = df_cumm.sc
    fig = plt.figure()
    ax = fig.add_axes([1,1,1,1])
    ax.plot(x, cc_values, label='Commercial Cards')
    ax.plot(x, rc_values, label='Retail Cards')
    ax.plot(x, sc_values, label='Student Cards')
    ax.set_xticks(x, [ arrow.get(str(m), "M").format("MMMM") for m in
df_cumm.index ], rotation=90)
    ax.set_xlabel("Month")
    ax.set_ylabel("Number of Calls")
    ax.set_title(f"Calls per Month for {y}")
```

```
ax.legend()
plt.show()
```



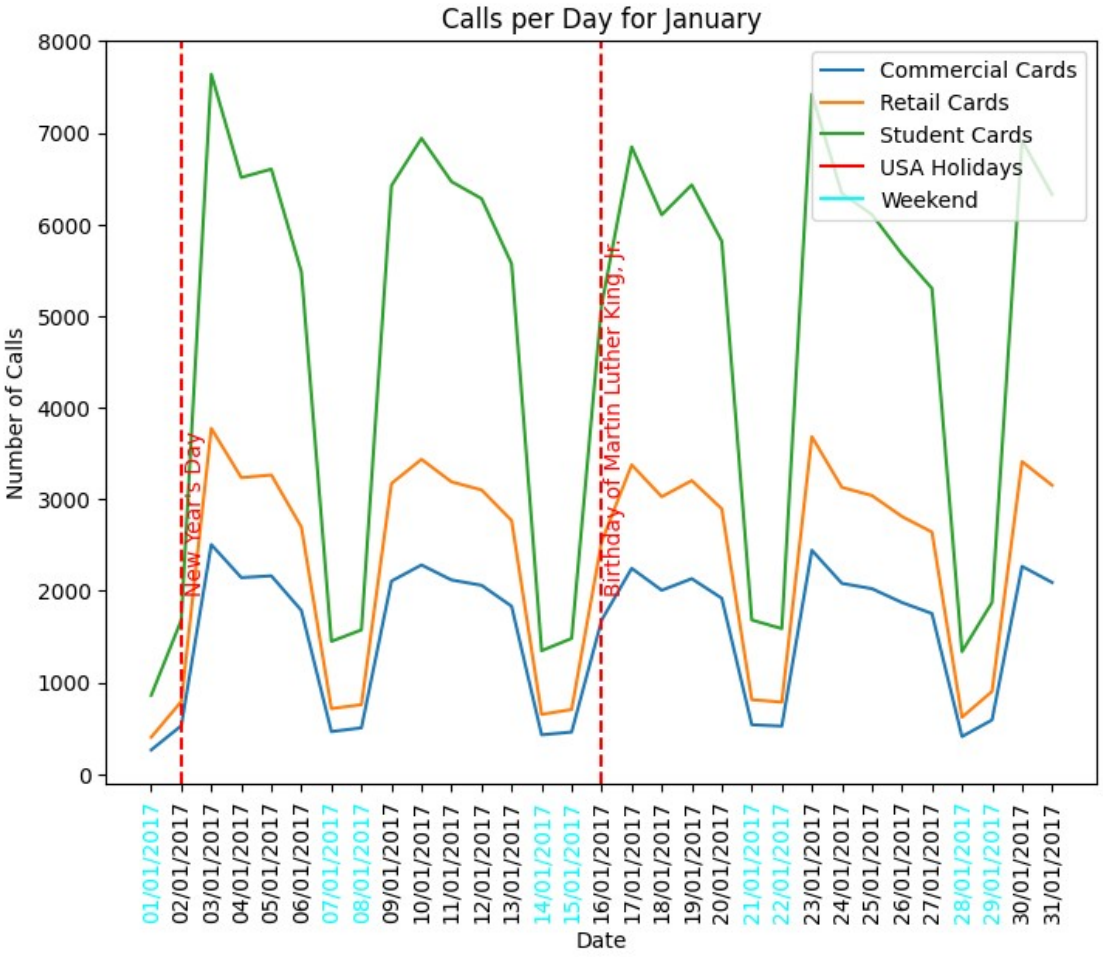

Calls per Month for 2019
160000
140000
120000
100000
80000
60000
40000
Number of Calls
Commercial Cards
Retail Cards
Student Cards
January
February
March
April
May
June
July
August
September
October
November
December
Month

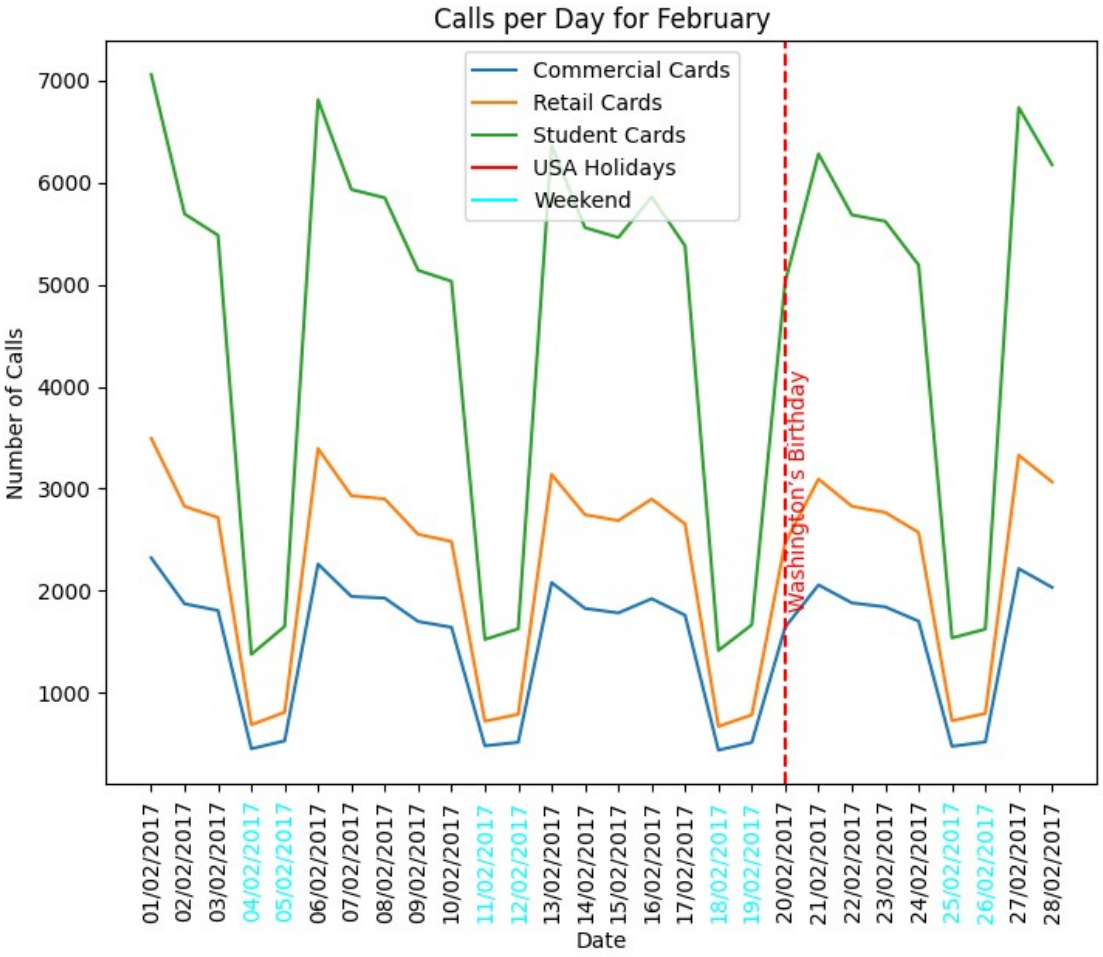

Calls per Month for 2020
Commercial Cards
Retail Cards
Student Cards
Number of Calls
140000
120000
100000
80000
60000
40000
January
February
March
April
May
June
July
August
September
October
November
December
Month

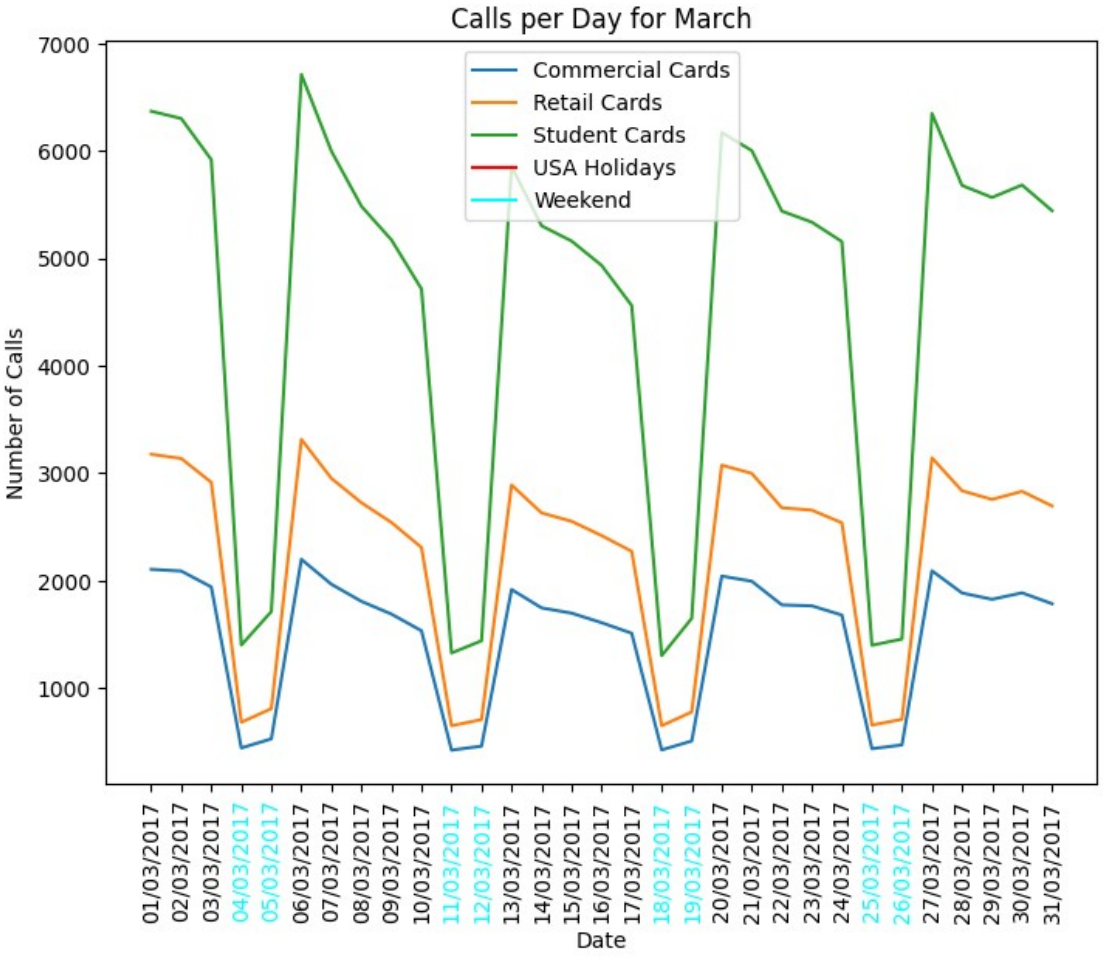


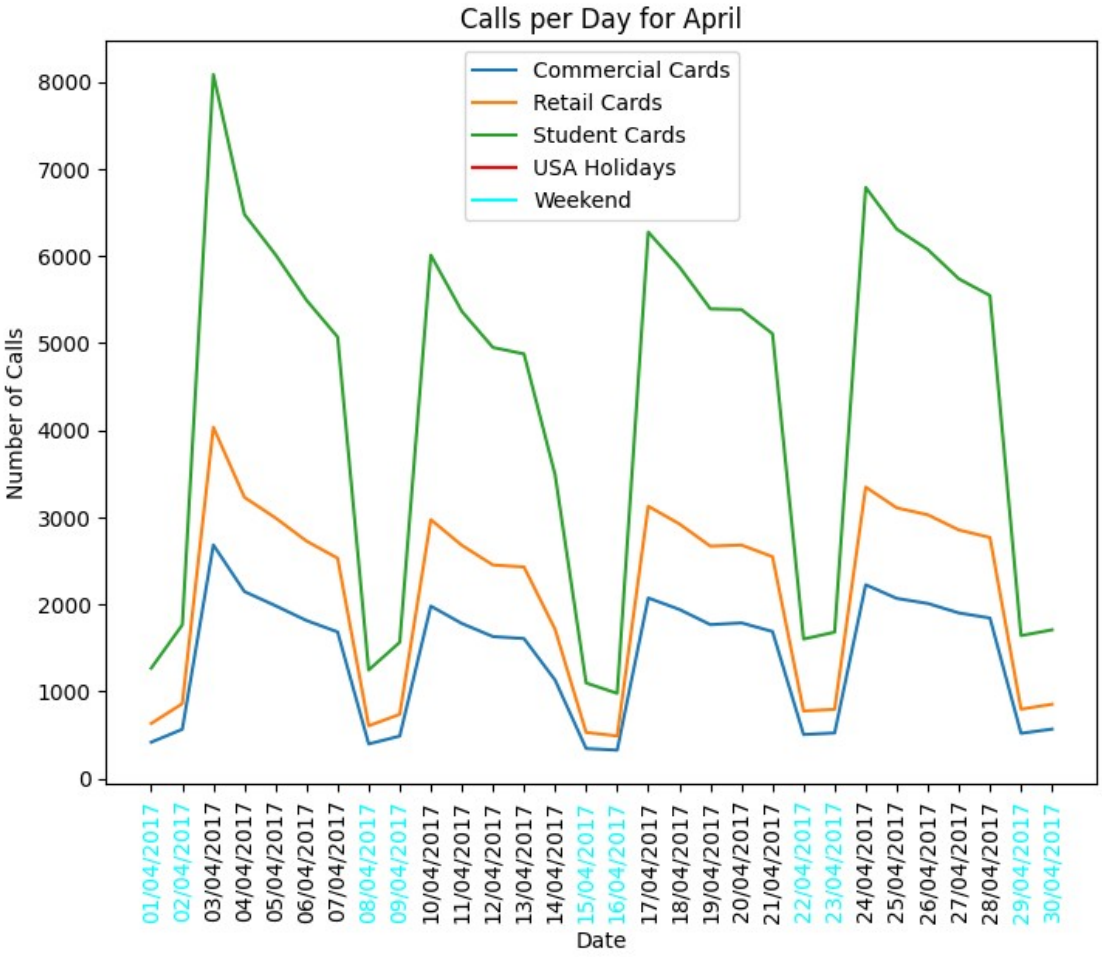

```
# Add US Fedrarl Holidays
cal = USFederalHolidayCalendar()
holidays_series = cal.holidays(start='2017-01-01', end='2021-04-30',
return_name=True)
holidays = holidays_series.to_frame("name")
holidays.reset_index(inplace=True)
holidays.rename(columns={"index":"date"}, inplace=True)

# 2017 data for each month
# Low number of calls on weekends
# Low number of calls on Big Holidays
holidays_2017 = holidays[[True if dt.year == 2017 else False for dt in
holidays.date]]
months = df_17.month.unique()
for m in months:
    x = range((df_17.month == m).sum())
    cc_values = df_17[df_17.month == m].cc
    rc_values = df_17[df_17.month == m].rc
    sc_values = df_17[df_17.month == m].sc
    month_holidays = holidays_2017[[True if dt.month == m else False
for dt in holidays_2017.date]]
    monhol_values = [(dt.date.day-1, dt['name']) for i, dt in
```

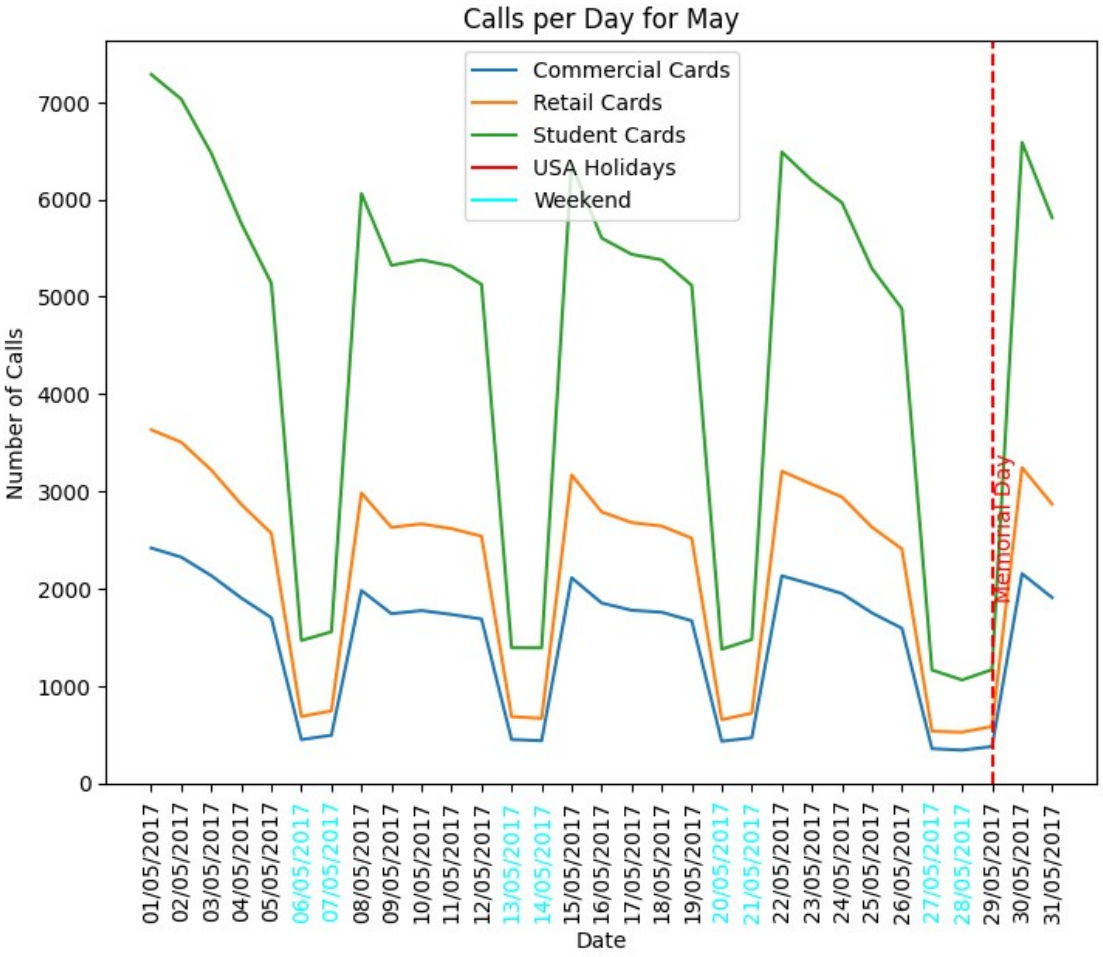
```
month_holidays.iterrows()]
    fig = plt.figure()
    ax = fig.add_axes([1,1,1,1])
    ax.plot(x, cc_values, label='Commercial Cards')
    ax.plot(x, rc_values, label='Retail Cards')
    ax.plot(x, sc_values, label='Student Cards')
    ax.plot([], [], label='USA Holidays', c='red')
    ax.plot([], [], label='Weekend', c='cyan')
    for mhd, mhn in monhol_values:
        if mhn == 'Thanksgiving Day':
            ax.axvline(mhd+1, c='red', ls='--')
            ax.text(mhd+1.1, ax.get_ylim()[1]//4, "Black Friday",
rotation=90, c='red')
        if mhn == 'Christmas Day':
            ax.axvline(mhd-1, c='red', ls='--')
            ax.text(mhd-0.9, ax.get_ylim()[1]//4, "Christman Eve",
rotation=90, c='red')
        ax.axvline(mhd, c='red', ls='--')
        ax.text(mhd+0.1, ax.get_ylim()[1]//4, mhn, rotation=90,
c='red')
    ax.set_xticks(x, df_17[df_17.month == m].date, rotation=90)
    for i, dt in enumerate(df_17[df_17.month == m].date):
        if arrow.get(dt, "DD/MM/YYYY").format("ddd") in ["Sat",
"Sun"]:
            ax.get_xticklabels()[i].set_color('cyan')
    ax.set_xlabel("Date")
    ax.set_ylabel("Number of Calls")
    ax.set_title(f"Calls per Day for {arrow.get(str(m),
'M').format('MMMM')}")
    ax.legend()
    plt.show()
```

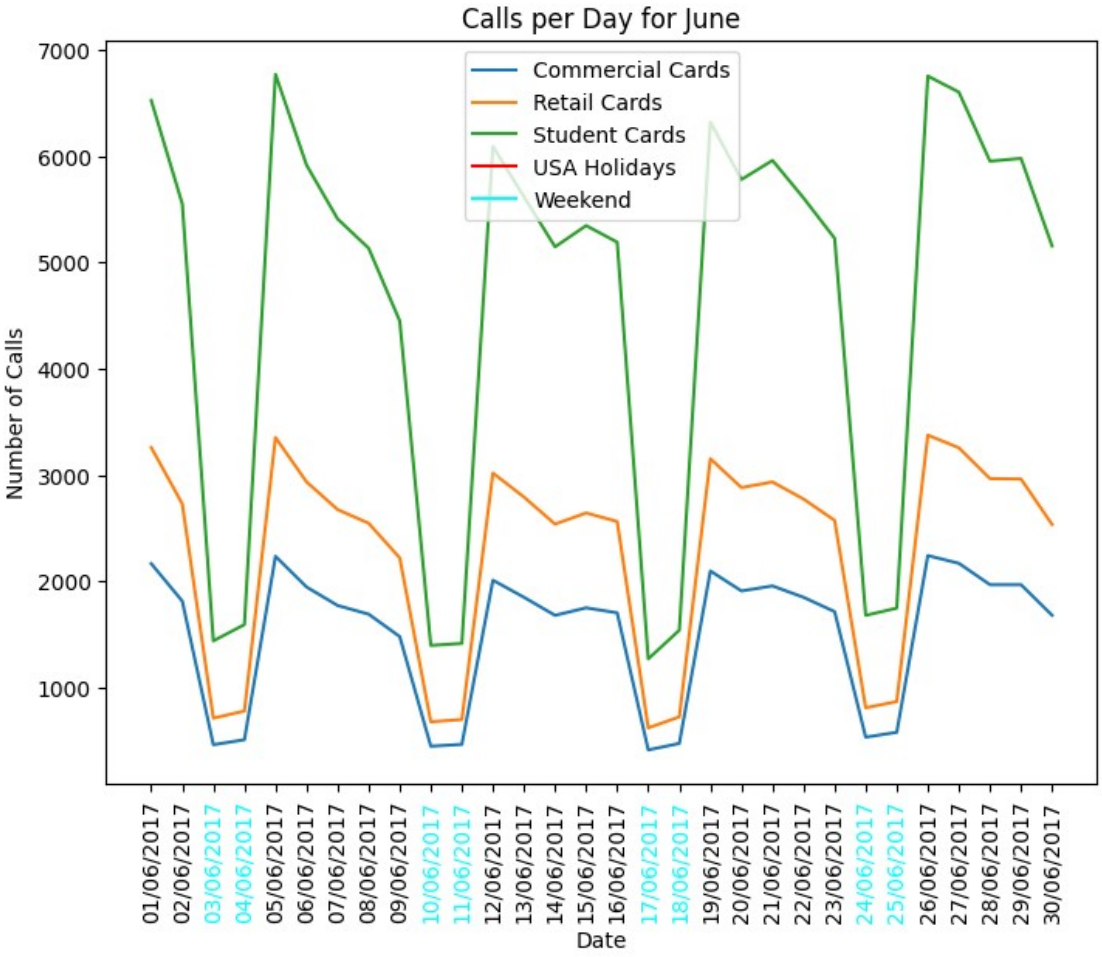

Calls per Day for January
Number of Calls
Date
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
New Year's Day
Birthday of Martin Luther King, Jr.
0
1000
2000
3000
4000
5000
6000
7000
8000
01/01/2017
02/01/2017
03/01/2017
04/01/2017
05/01/2017
06/01/2017
07/01/2017
08/01/2017
09/01/2017
10/01/2017
11/01/2017
12/01/2017
13/01/2017
14/01/2017
15/01/2017
16/01/2017
17/01/2017
18/01/2017
19/01/2017
20/01/2017
21/01/2017
22/01/2017
23/01/2017
24/01/2017
25/01/2017
26/01/2017
27/01/2017
28/01/2017
29/01/2017
30/01/2017
31/01/2017

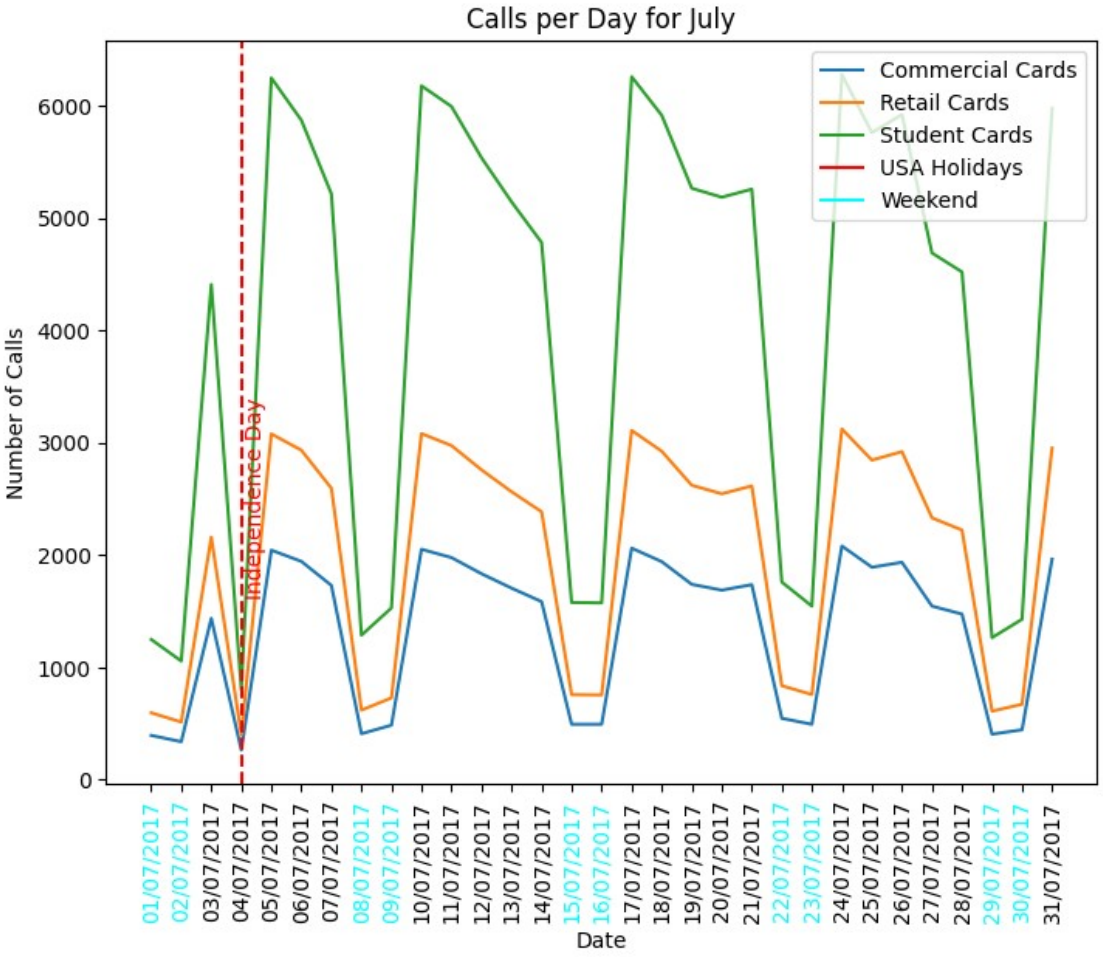

Calls per Day for February
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
7000
6000
5000
4000
3000
2000
1000
Washington's Birthday
01/02/2017
02/02/2017
03/02/2017
04/02/2017
05/02/2017
06/02/2017
07/02/2017
08/02/2017
09/02/2017
10/02/2017
11/02/2017
12/02/2017
13/02/2017
14/02/2017
15/02/2017
16/02/2017
17/02/2017
18/02/2017
19/02/2017
20/02/2017
21/02/2017
22/02/2017
23/02/2017
24/02/2017
25/02/2017
26/02/2017
27/02/2017
28/02/2017
Date

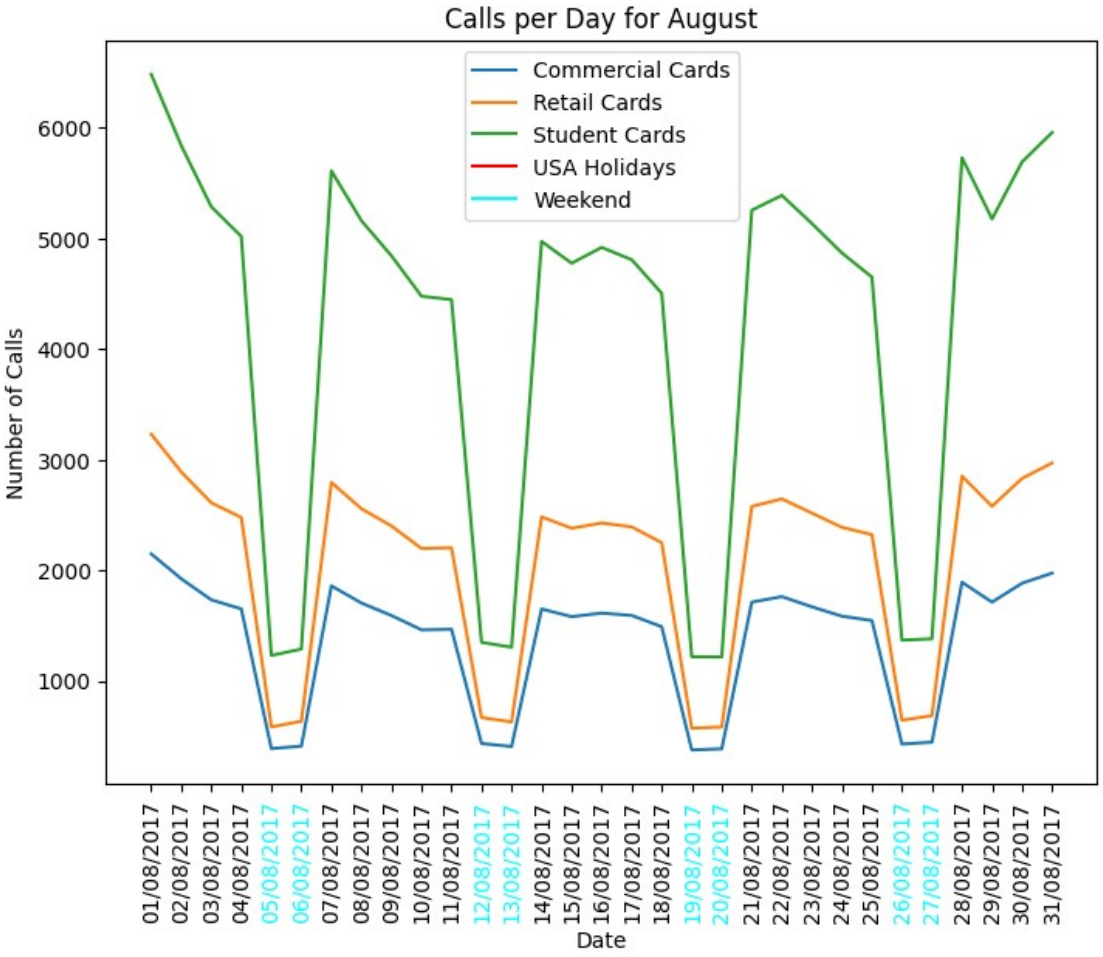

Calls per Day for March
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
7000
6000
5000
4000
3000
2000
1000
01/03/2017
02/03/2017
03/03/2017
04/03/2017
05/03/2017
06/03/2017
07/03/2017
08/03/2017
09/03/2017
10/03/2017
11/03/2017
12/03/2017
13/03/2017
14/03/2017
15/03/2017
16/03/2017
17/03/2017
18/03/2017
19/03/2017
20/03/2017
21/03/2017
22/03/2017
23/03/2017
24/03/2017
25/03/2017
26/03/2017
27/03/2017
28/03/2017
29/03/2017
30/03/2017
31/03/2017
Date

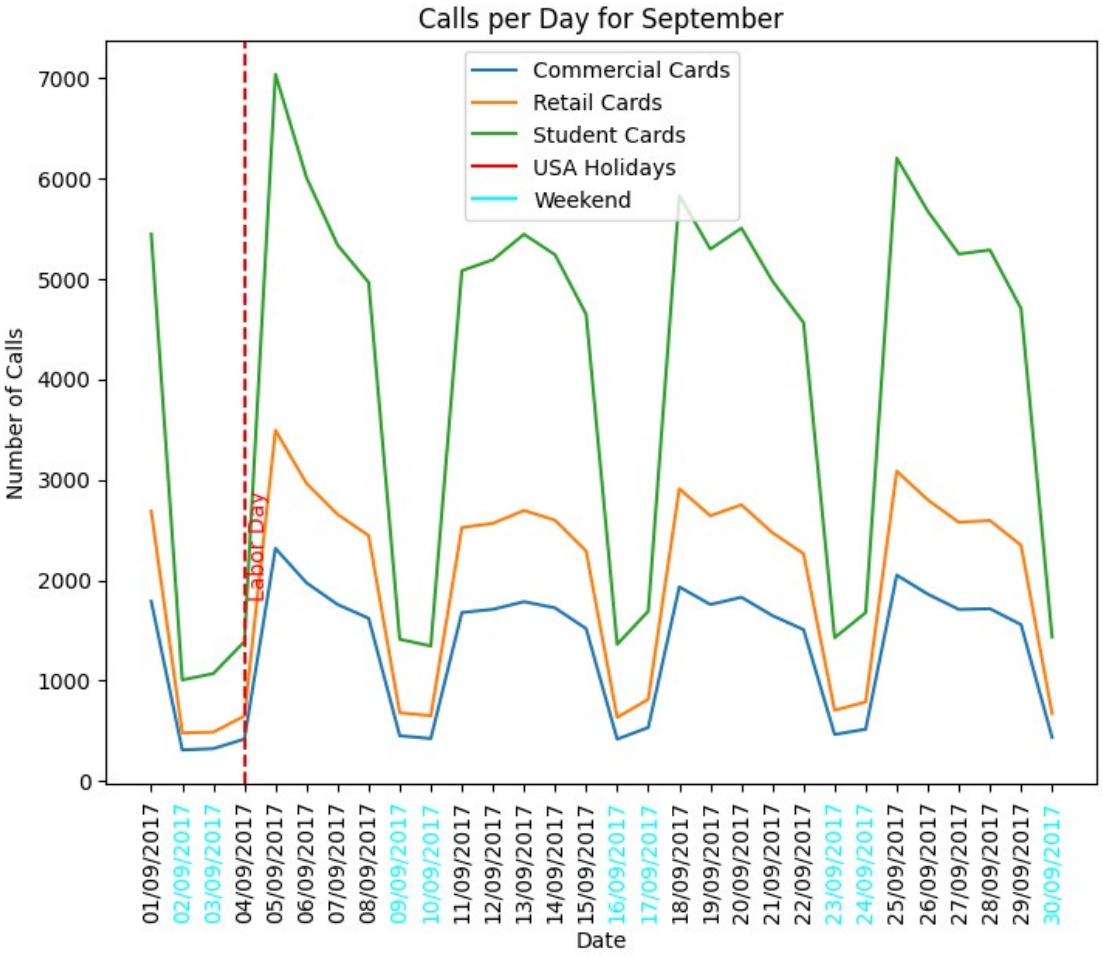

Calls per Day for April
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
8000
01/04/2017
02/04/2017
03/04/2017
04/04/2017
05/04/2017
06/04/2017
07/04/2017
08/04/2017
09/04/2017
10/04/2017
11/04/2017
12/04/2017
13/04/2017
14/04/2017
15/04/2017
16/04/2017
17/04/2017
18/04/2017
19/04/2017
20/04/2017
21/04/2017
22/04/2017
23/04/2017
24/04/2017
25/04/2017
26/04/2017
27/04/2017
28/04/2017
29/04/2017
30/04/2017
Date

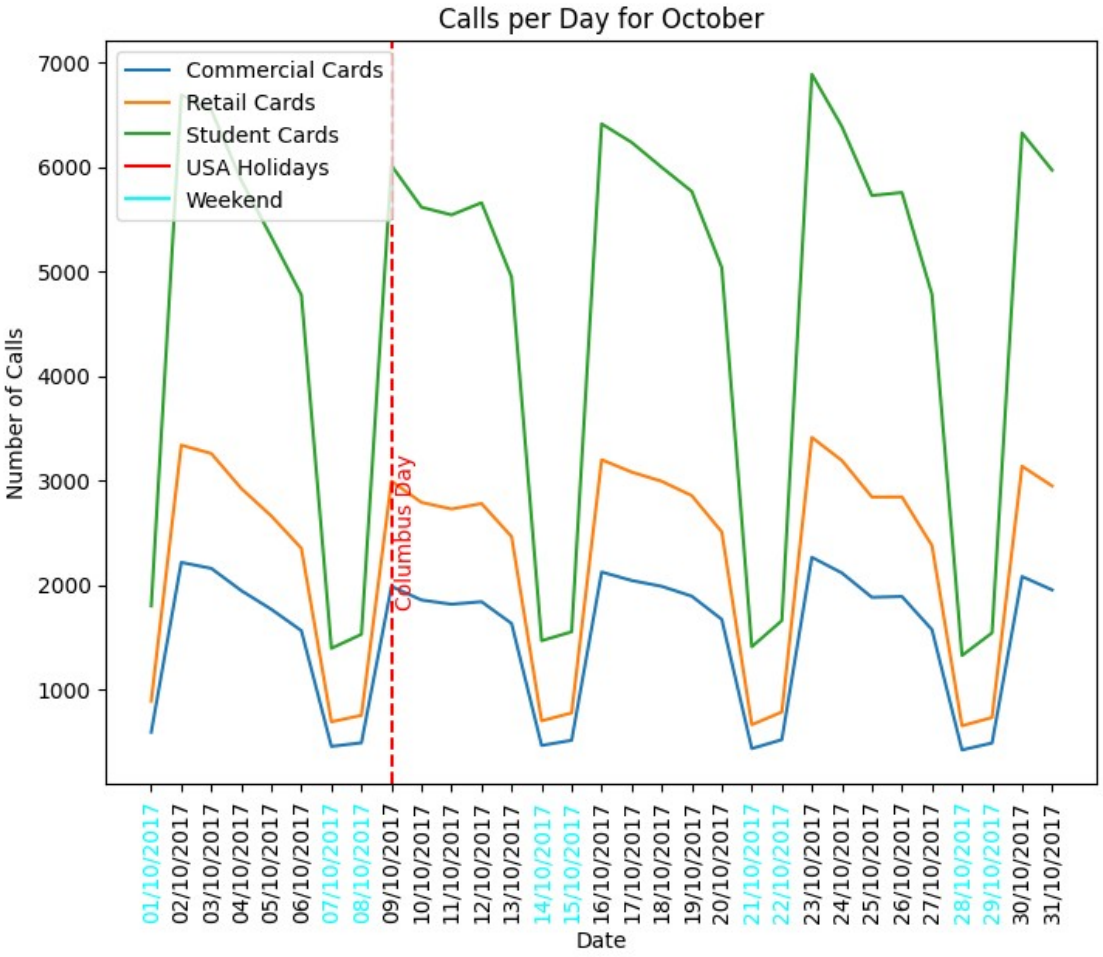

Calls per Day for May
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
Memorial Day
01/05/2017
02/05/2017
03/05/2017
04/05/2017
05/05/2017
06/05/2017
07/05/2017
08/05/2017
09/05/2017
10/05/2017
11/05/2017
12/05/2017
13/05/2017
14/05/2017
15/05/2017
16/05/2017
17/05/2017
18/05/2017
19/05/2017
20/05/2017
21/05/2017
22/05/2017
23/05/2017
24/05/2017
25/05/2017
26/05/2017
27/05/2017
28/05/2017
29/05/2017
30/05/2017
31/05/2017
Date

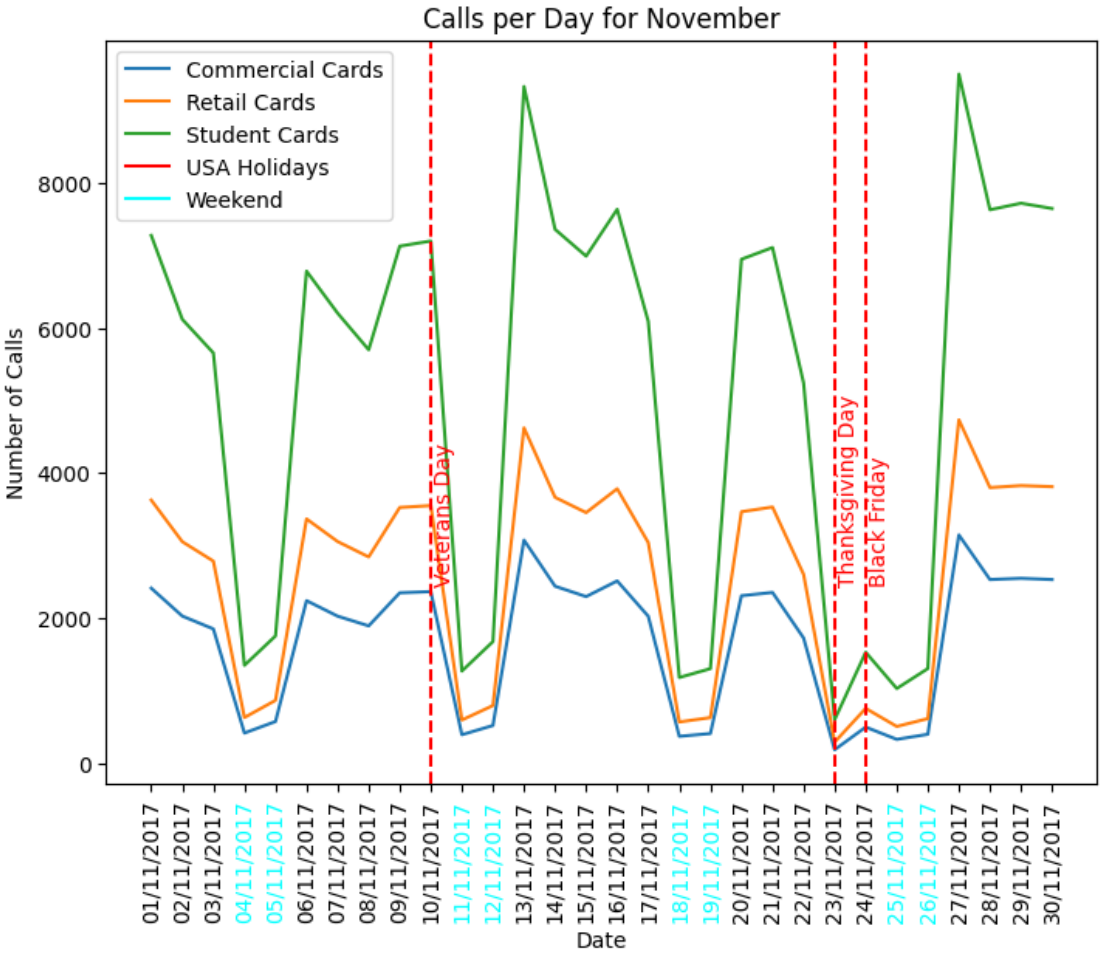

Calls per Day for June
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
7000
6000
5000
4000
3000
2000
1000
01/06/2017
02/06/2017
03/06/2017
04/06/2017
05/06/2017
06/06/2017
07/06/2017
08/06/2017
09/06/2017
10/06/2017
11/06/2017
12/06/2017
13/06/2017
14/06/2017
15/06/2017
16/06/2017
17/06/2017
18/06/2017
19/06/2017
20/06/2017
21/06/2017
22/06/2017
23/06/2017
24/06/2017
25/06/2017
26/06/2017
27/06/2017
28/06/2017
29/06/2017
30/06/2017
Date

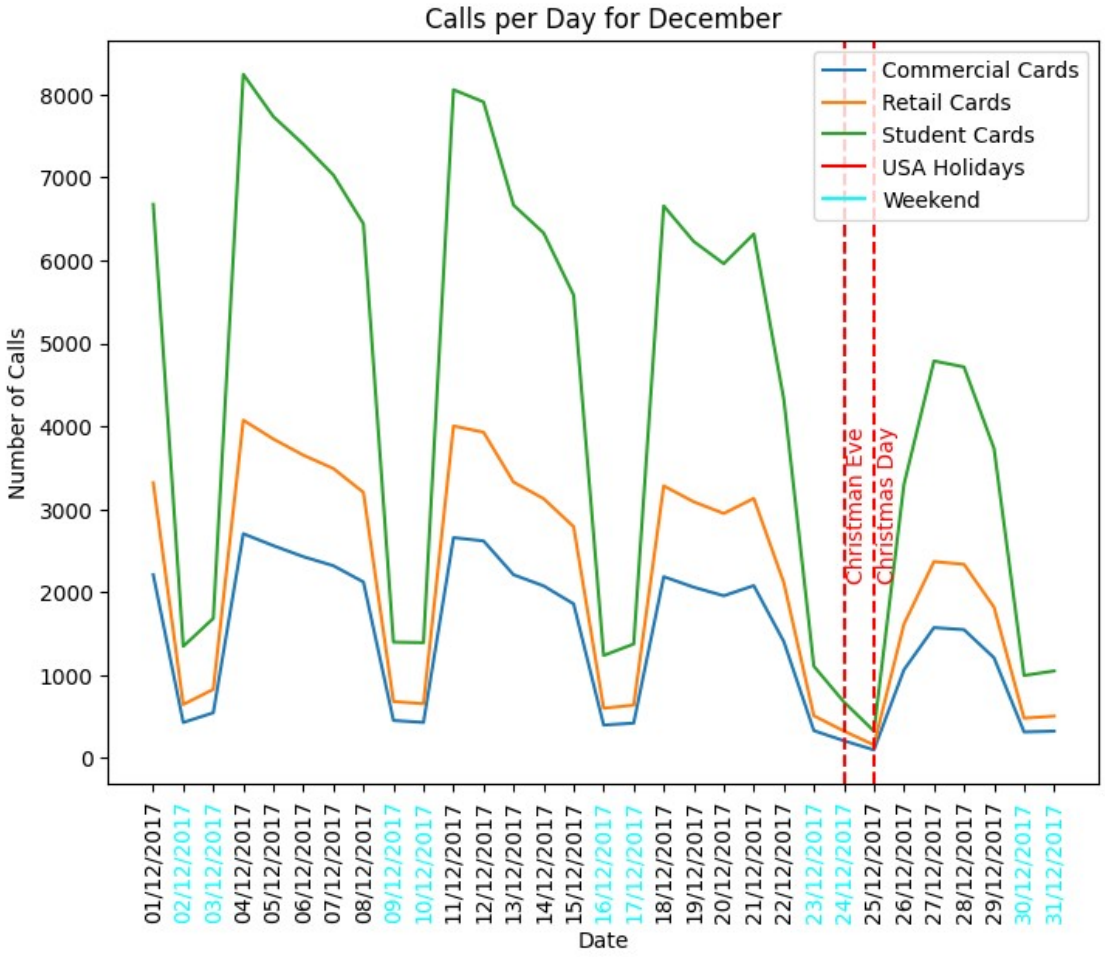

Calls per Day for July
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
Independence Day
01/07/2017
02/07/2017
03/07/2017
04/07/2017
05/07/2017
06/07/2017
07/07/2017
08/07/2017
09/07/2017
10/07/2017
11/07/2017
12/07/2017
13/07/2017
14/07/2017
15/07/2017
16/07/2017
17/07/2017
18/07/2017
19/07/2017
20/07/2017
21/07/2017
22/07/2017
23/07/2017
24/07/2017
25/07/2017
26/07/2017
27/07/2017
28/07/2017
29/07/2017
30/07/2017
31/07/2017
Date


Calls per Day for August
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
6000
5000
4000
3000
2000
1000
01/08/2017
02/08/2017
03/08/2017
04/08/2017
05/08/2017
06/08/2017
07/08/2017
08/08/2017
09/08/2017
10/08/2017
11/08/2017
12/08/2017
13/08/2017
14/08/2017
15/08/2017
16/08/2017
17/08/2017
18/08/2017
19/08/2017
20/08/2017
21/08/2017
22/08/2017
23/08/2017
24/08/2017
25/08/2017
26/08/2017
27/08/2017
28/08/2017
29/08/2017
30/08/2017
31/08/2017
Date


Calls per Day for September
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
Labor Day
Date

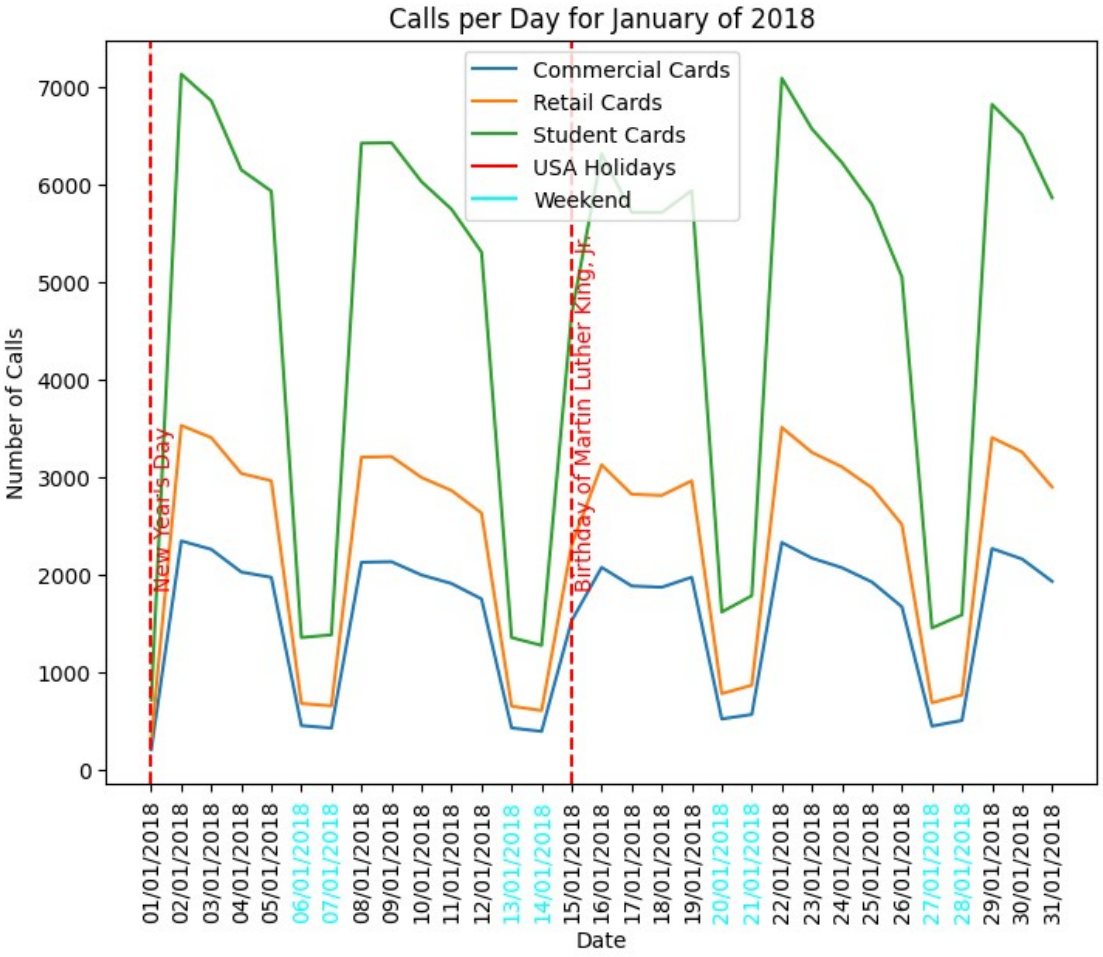

Calls per Day for October
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
7000
6000
5000
4000
3000
2000
1000
Columbus Day
01/10/2017
02/10/2017
03/10/2017
04/10/2017
05/10/2017
06/10/2017
07/10/2017
08/10/2017
09/10/2017
10/10/2017
11/10/2017
12/10/2017
13/10/2017
14/10/2017
15/10/2017
16/10/2017
17/10/2017
18/10/2017
19/10/2017
20/10/2017
21/10/2017
22/10/2017
23/10/2017
24/10/2017
25/10/2017
26/10/2017
27/10/2017
28/10/2017
29/10/2017
30/10/2017
31/10/2017
Date

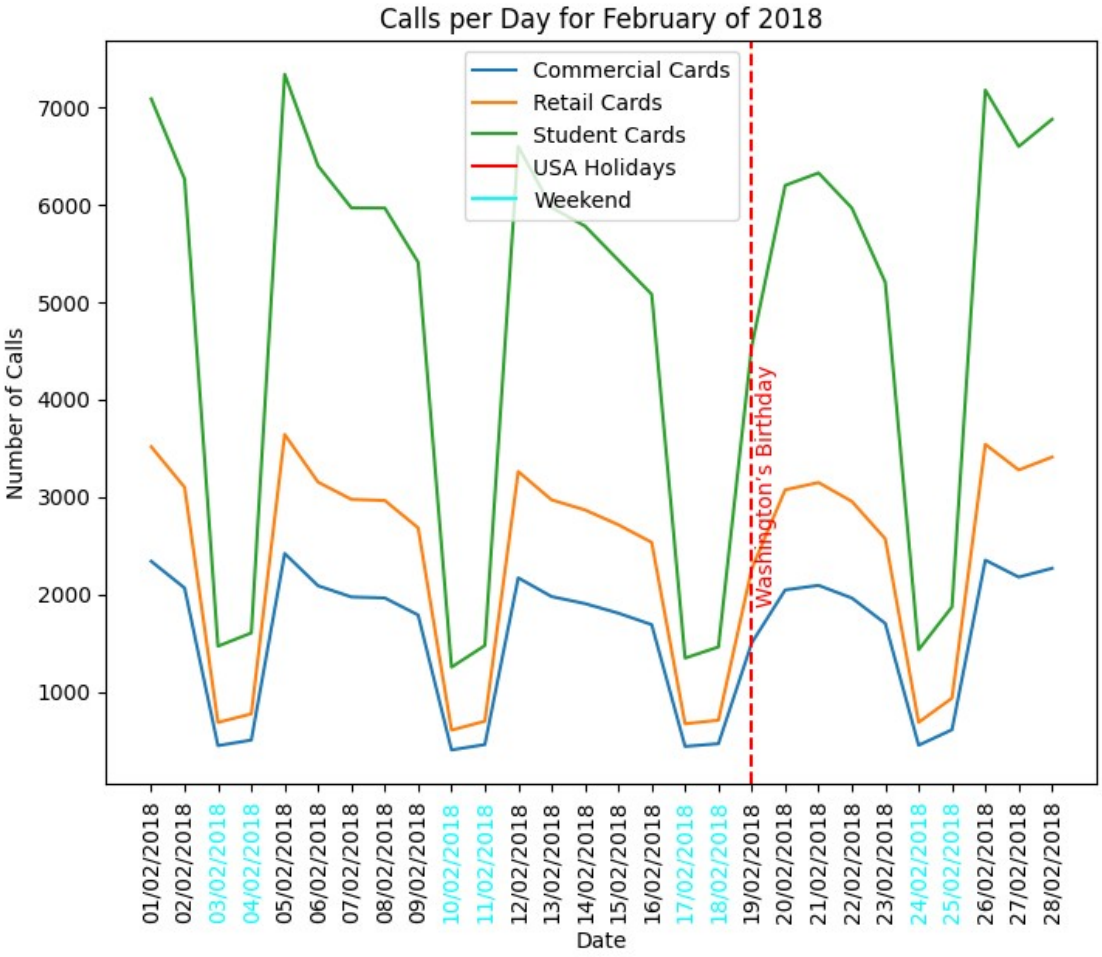

Calls per Day for November
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
2000
4000
6000
8000
Veterans Day
Thanksgiving Day
Black Friday
01/11/2017
02/11/2017
03/11/2017
04/11/2017
05/11/2017
06/11/2017
07/11/2017
08/11/2017
09/11/2017
10/11/2017
11/11/2017
12/11/2017
13/11/2017
14/11/2017
15/11/2017
16/11/2017
17/11/2017
18/11/2017
19/11/2017
20/11/2017
21/11/2017
22/11/2017
23/11/2017
24/11/2017
25/11/2017
26/11/2017
27/11/2017
28/11/2017
29/11/2017
30/11/2017
Date

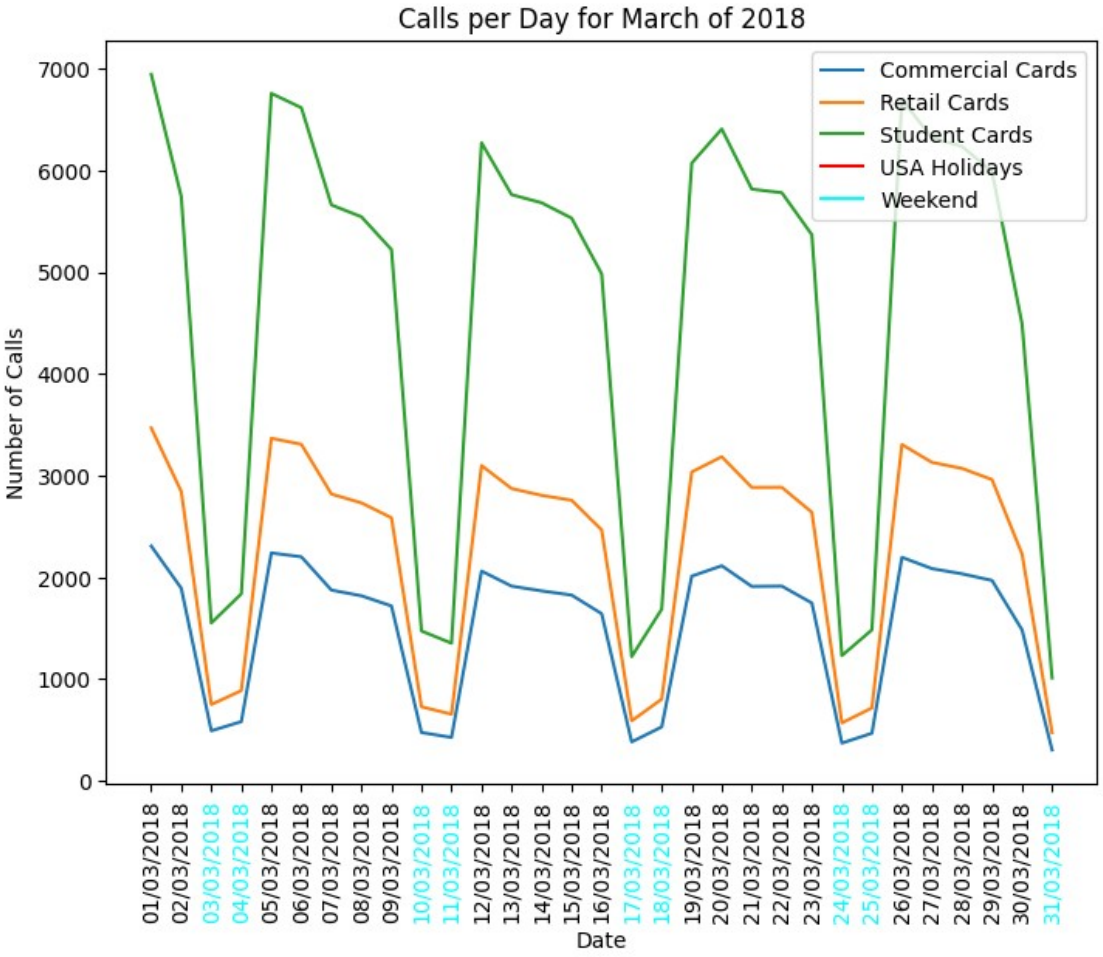


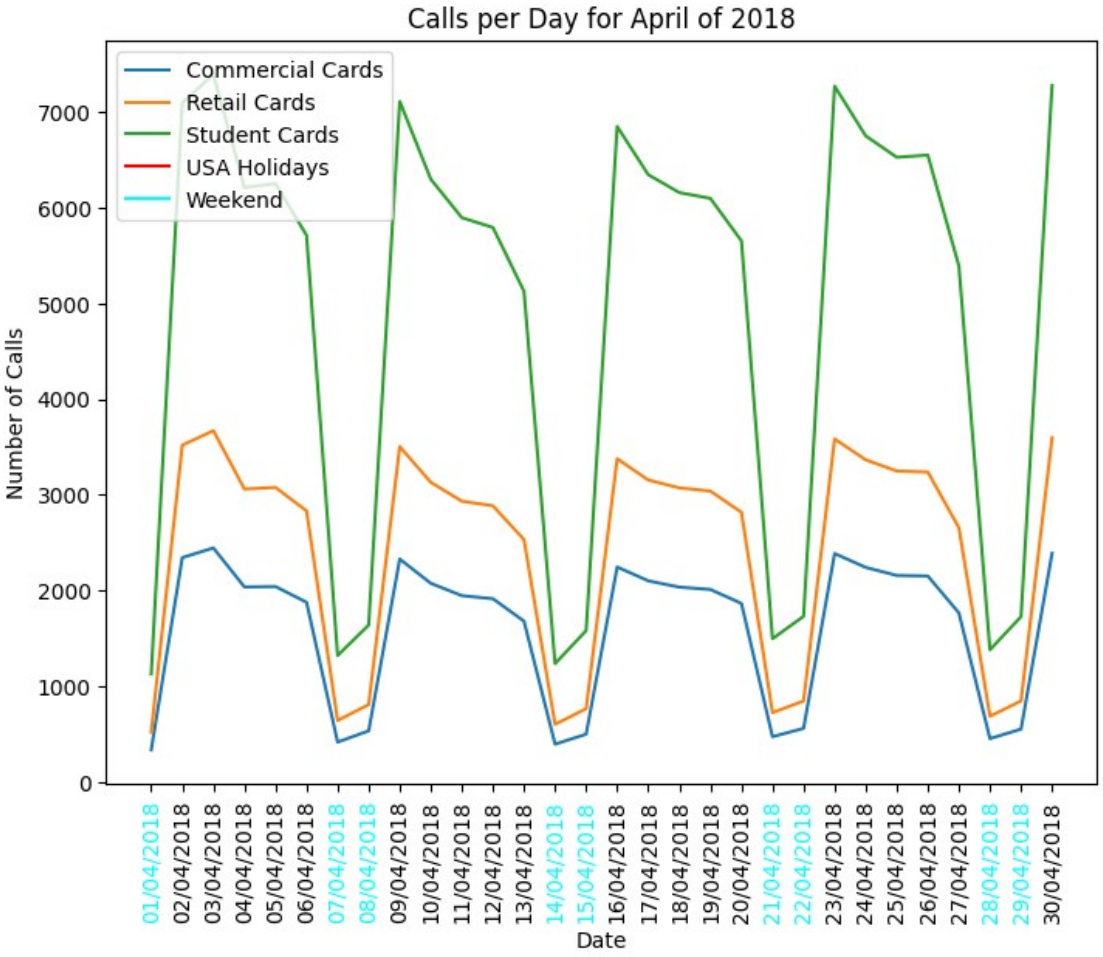

```
# 2018-2021 data for each month
# Low number of calls on weekends
# Low number of calls on Big Holidays
years = df_not17_sorted.year.unique()
for y in years:
    holidays_yearly = holidays[[True if dt.year == y else False for dt
in holidays.date]]
    df_not17_sorted_yearly = df_not17_sorted[df_not17_sorted.year ==
y]
    months = df_not17_sorted_yearly.month.unique()
    for m in months:
        x = range((df_not17_sorted_yearly.month == m).sum())
        cc_values =
df_not17_sorted_yearly[df_not17_sorted_yearly.month == m].cc
        rc_values =
df_not17_sorted_yearly[df_not17_sorted_yearly.month == m].rc
        sc_values =
df_not17_sorted_yearly[df_not17_sorted_yearly.month == m].sc
        month_holidays = holidays_yearly[[True if dt.month == m else
False for dt in holidays_yearly.date]]
        monhol_values = [(dt.date.day-1, dt['name']) for i, dt in
```

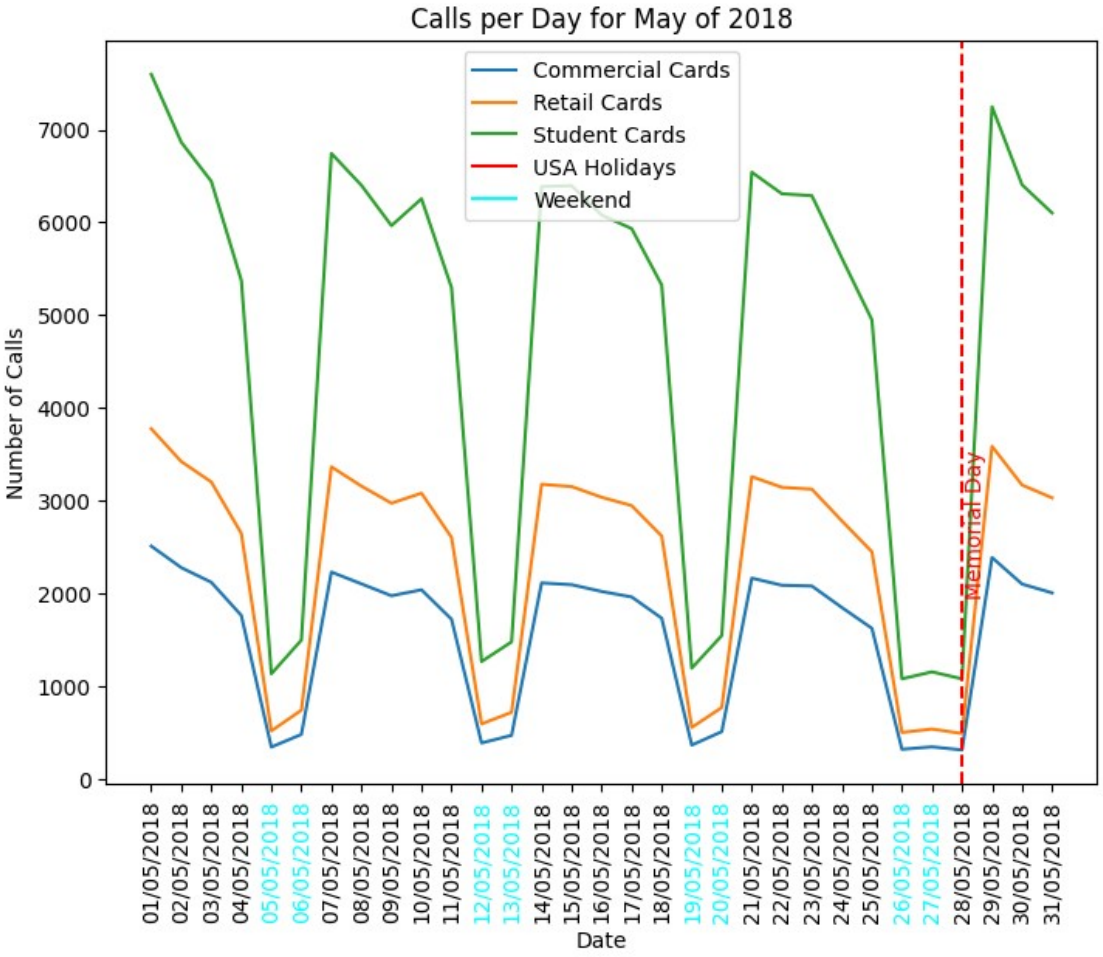
```
month_holidays.iterrows()]
        fig = plt.figure()
        ax = fig.add_axes([1,1,1,1])
        ax.plot(x, cc_values, label='Commercial Cards')
        ax.plot(x, rc_values, label='Retail Cards')
        ax.plot(x, sc_values, label='Student Cards')
        ax.plot([], [], label='USA Holidays', c='red')
        ax.plot([], [], label='Weekend', c='cyan')
        for mhd, mhn in monhol_values:
            if mhn == 'Thanksgiving Day':
                ax.axvline(mhd+1, c='red', ls='--')
                ax.text(mhd+1.1, ax.get_ylim()[1]//4, "Black Friday",
rotation=90, c='red')
            if mhn == 'Christmas Day':
                ax.axvline(mhd-1, c='red', ls='--')
                ax.text(mhd-0.9, ax.get_ylim()[1]//4, "Christman Eve",
rotation=90, c='red')
            ax.axvline(mhd, c='red', ls='--')
            ax.text(mhd+0.1, ax.get_ylim()[1]//4, mhn, rotation=90,
c='red')
        ax.set_xticks(x,
df_not17_sorted_yearly[df_not17_sorted_yearly.month == m].date,
rotation=90)
        for i, dt in
enumerate(df_not17_sorted_yearly[df_not17_sorted_yearly.month ==
m].date):
            if arrow.get(dt, "DD/MM/YYYY").format("ddd") in ["Sat",
"Sun"]:
                ax.get_xticklabels()[i].set_color('cyan')
        ax.set_xlabel("Date")
        ax.set_ylabel("Number of Calls")
        ax.set_title(f"Calls per Day for {arrow.get(str(m),
'M').format('MMMM')} of {y}")
        ax.legend()
        plt.show()
```

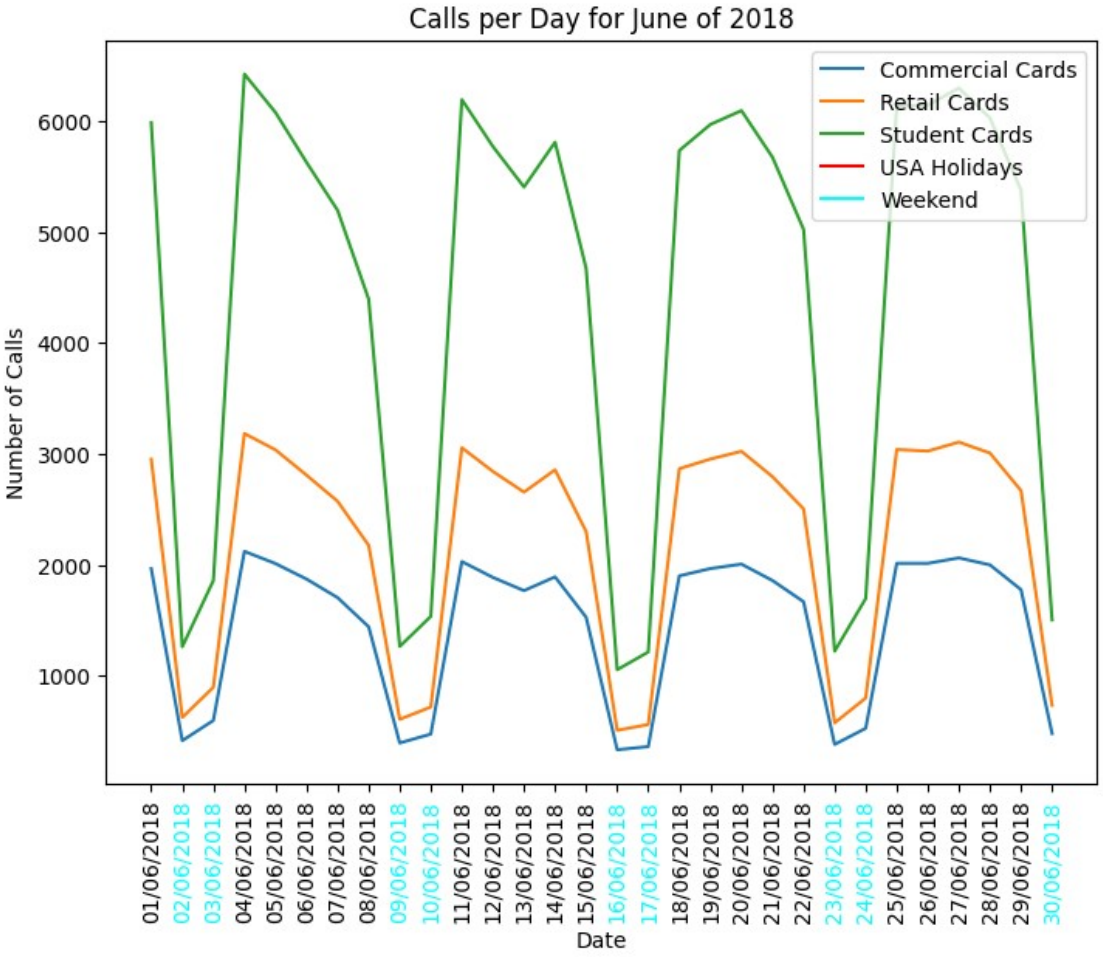

Calls per Day for January of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
New Year's Day
Birthday of Martin Luther King, Jr.
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/01/2018
02/01/2018
03/01/2018
04/01/2018
05/01/2018
06/01/2018
07/01/2018
08/01/2018
09/01/2018
10/01/2018
11/01/2018
12/01/2018
13/01/2018
14/01/2018
15/01/2018
16/01/2018
17/01/2018
18/01/2018
19/01/2018
20/01/2018
21/01/2018
22/01/2018
23/01/2018
24/01/2018
25/01/2018
26/01/2018
27/01/2018
28/01/2018
29/01/2018
30/01/2018
31/01/2018
Date

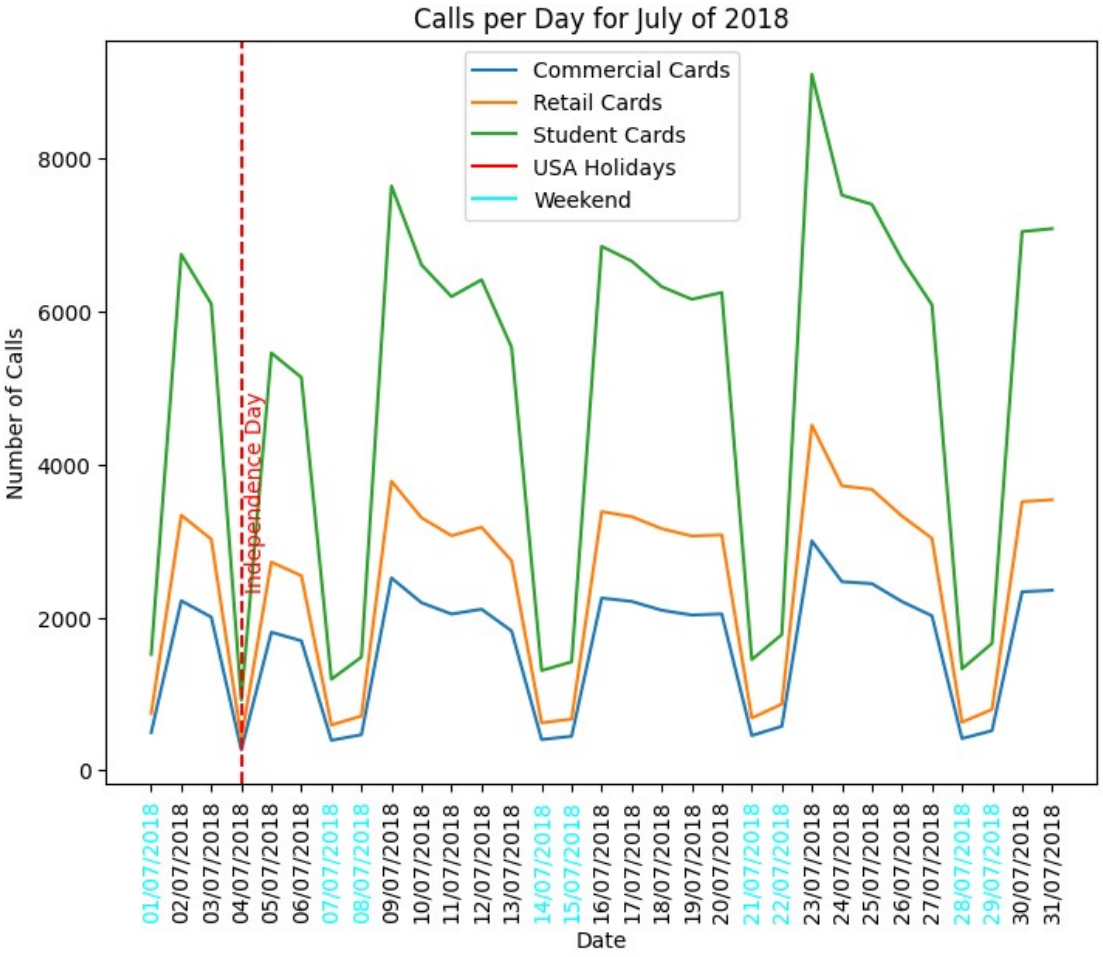

Calls per Day for February of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
7000
6000
5000
4000
3000
2000
1000
Washington's Birthday
01/02/2018
02/02/2018
03/02/2018
04/02/2018
05/02/2018
06/02/2018
07/02/2018
08/02/2018
09/02/2018
10/02/2018
11/02/2018
12/02/2018
13/02/2018
14/02/2018
15/02/2018
16/02/2018
17/02/2018
18/02/2018
19/02/2018
20/02/2018
21/02/2018
22/02/2018
23/02/2018
24/02/2018
25/02/2018
26/02/2018
27/02/2018
28/02/2018
Date

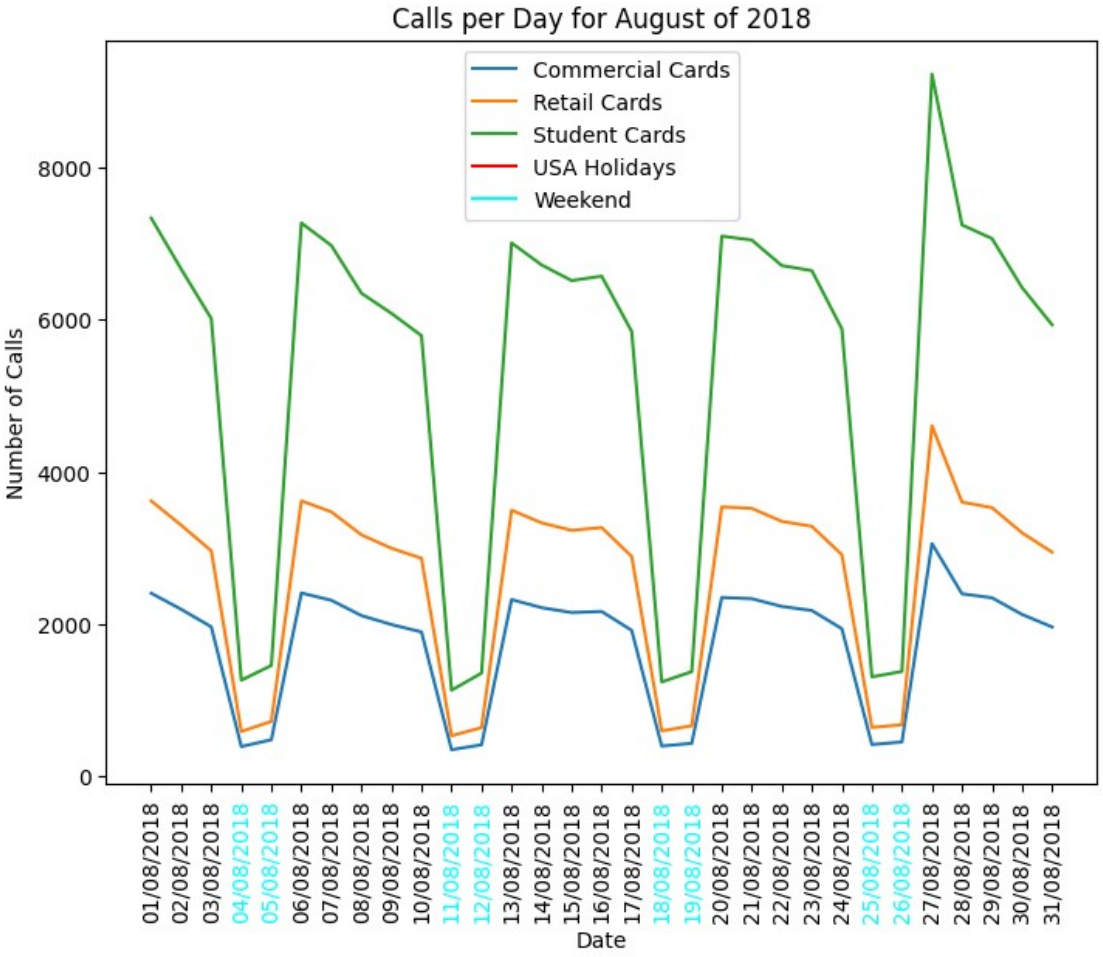

Calls per Day for March of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/03/2018
02/03/2018
03/03/2018
04/03/2018
05/03/2018
06/03/2018
07/03/2018
08/03/2018
09/03/2018
10/03/2018
11/03/2018
12/03/2018
13/03/2018
14/03/2018
15/03/2018
16/03/2018
17/03/2018
18/03/2018
19/03/2018
20/03/2018
21/03/2018
22/03/2018
23/03/2018
24/03/2018
25/03/2018
26/03/2018
27/03/2018
28/03/2018
29/03/2018
30/03/2018
31/03/2018
Date

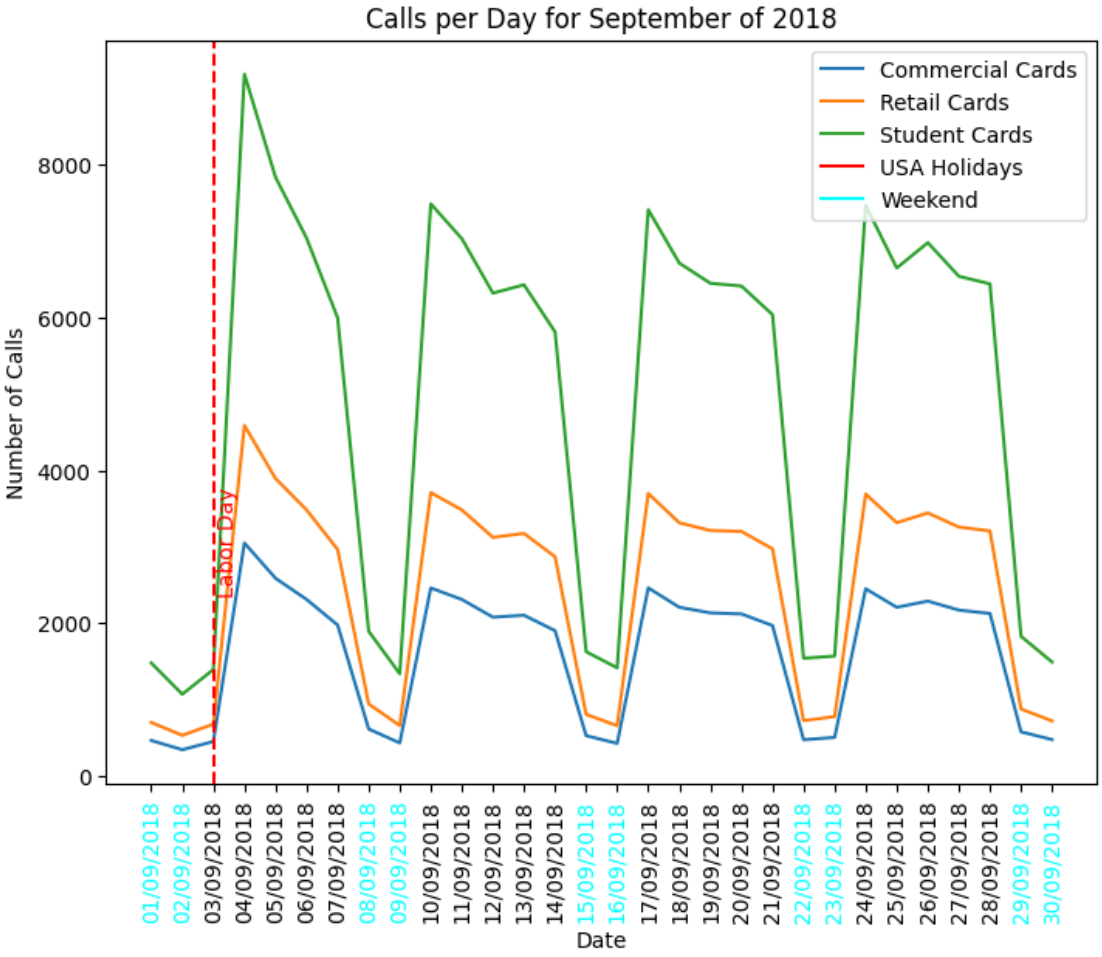

Calls per Day for April of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/04/2018
02/04/2018
03/04/2018
04/04/2018
05/04/2018
06/04/2018
07/04/2018
08/04/2018
09/04/2018
10/04/2018
11/04/2018
12/04/2018
13/04/2018
14/04/2018
15/04/2018
16/04/2018
17/04/2018
18/04/2018
19/04/2018
20/04/2018
21/04/2018
22/04/2018
23/04/2018
24/04/2018
25/04/2018
26/04/2018
27/04/2018
28/04/2018
29/04/2018
30/04/2018
Date

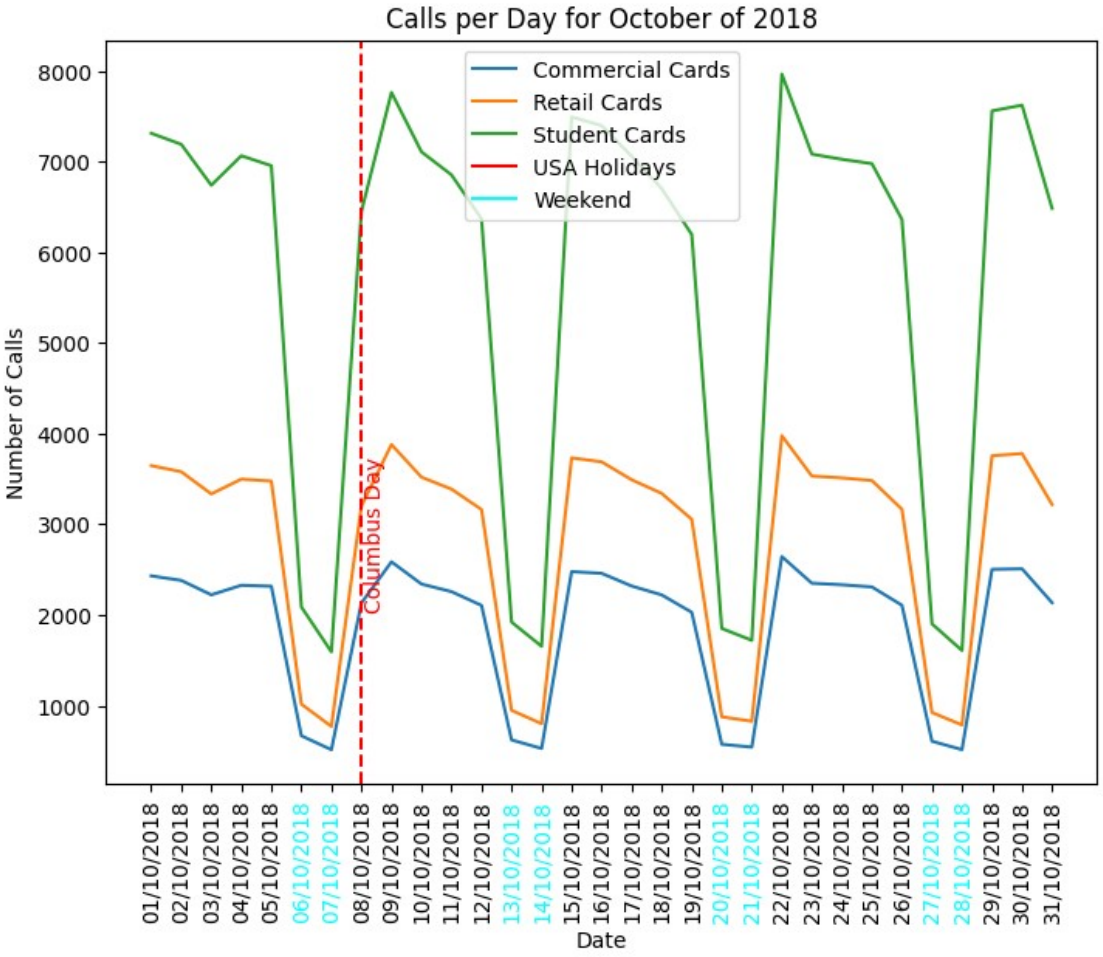

Calls per Day for May of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Memorial Day
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/05/2018
02/05/2018
03/05/2018
04/05/2018
05/05/2018
06/05/2018
07/05/2018
08/05/2018
09/05/2018
10/05/2018
11/05/2018
12/05/2018
13/05/2018
14/05/2018
15/05/2018
16/05/2018
17/05/2018
18/05/2018
19/05/2018
20/05/2018
21/05/2018
22/05/2018
23/05/2018
24/05/2018
25/05/2018
26/05/2018
27/05/2018
28/05/2018
29/05/2018
30/05/2018
31/05/2018
Date

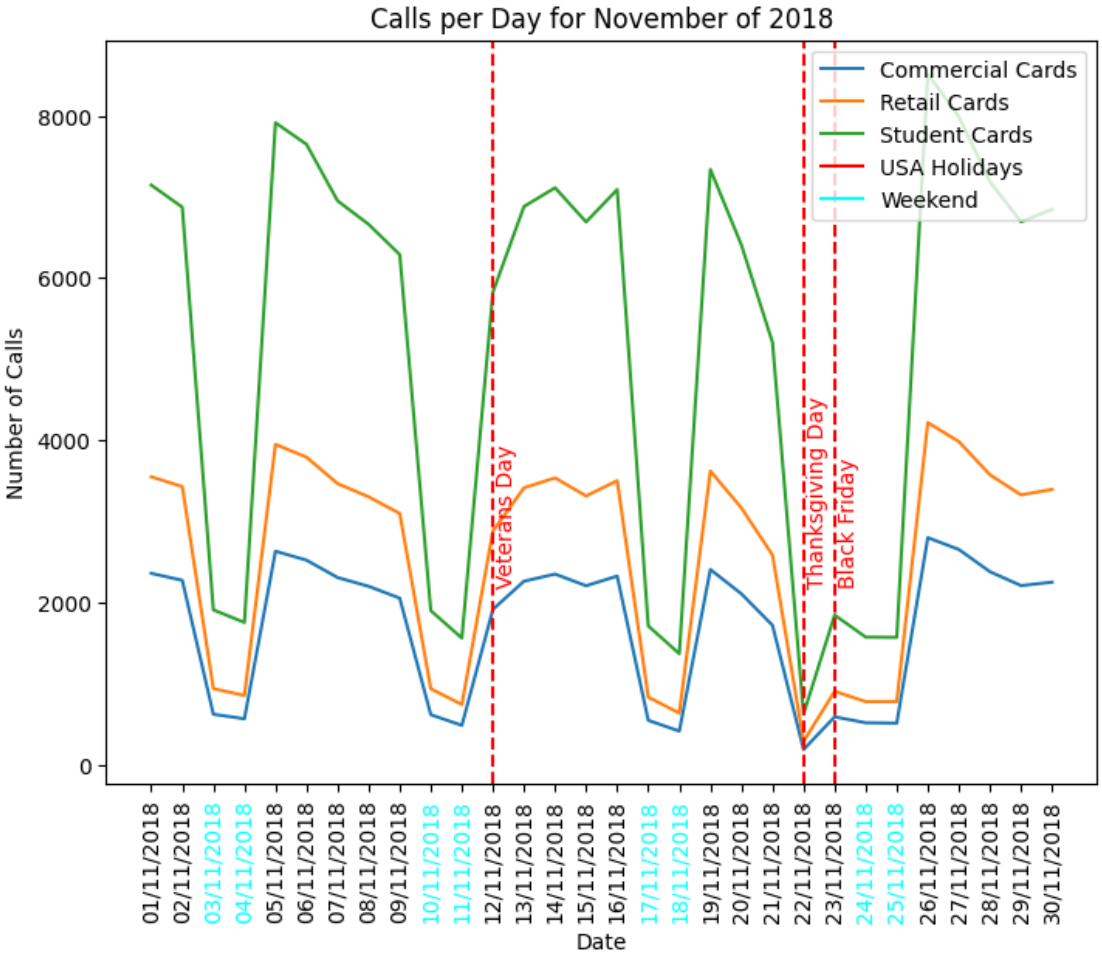

Calls per Day for June of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
6000
5000
4000
3000
2000
1000
01/06/2018
02/06/2018
03/06/2018
04/06/2018
05/06/2018
06/06/2018
07/06/2018
08/06/2018
09/06/2018
10/06/2018
11/06/2018
12/06/2018
13/06/2018
14/06/2018
15/06/2018
16/06/2018
17/06/2018
18/06/2018
19/06/2018
20/06/2018
21/06/2018
22/06/2018
23/06/2018
24/06/2018
25/06/2018
26/06/2018
27/06/2018
28/06/2018
29/06/2018
30/06/2018
Date

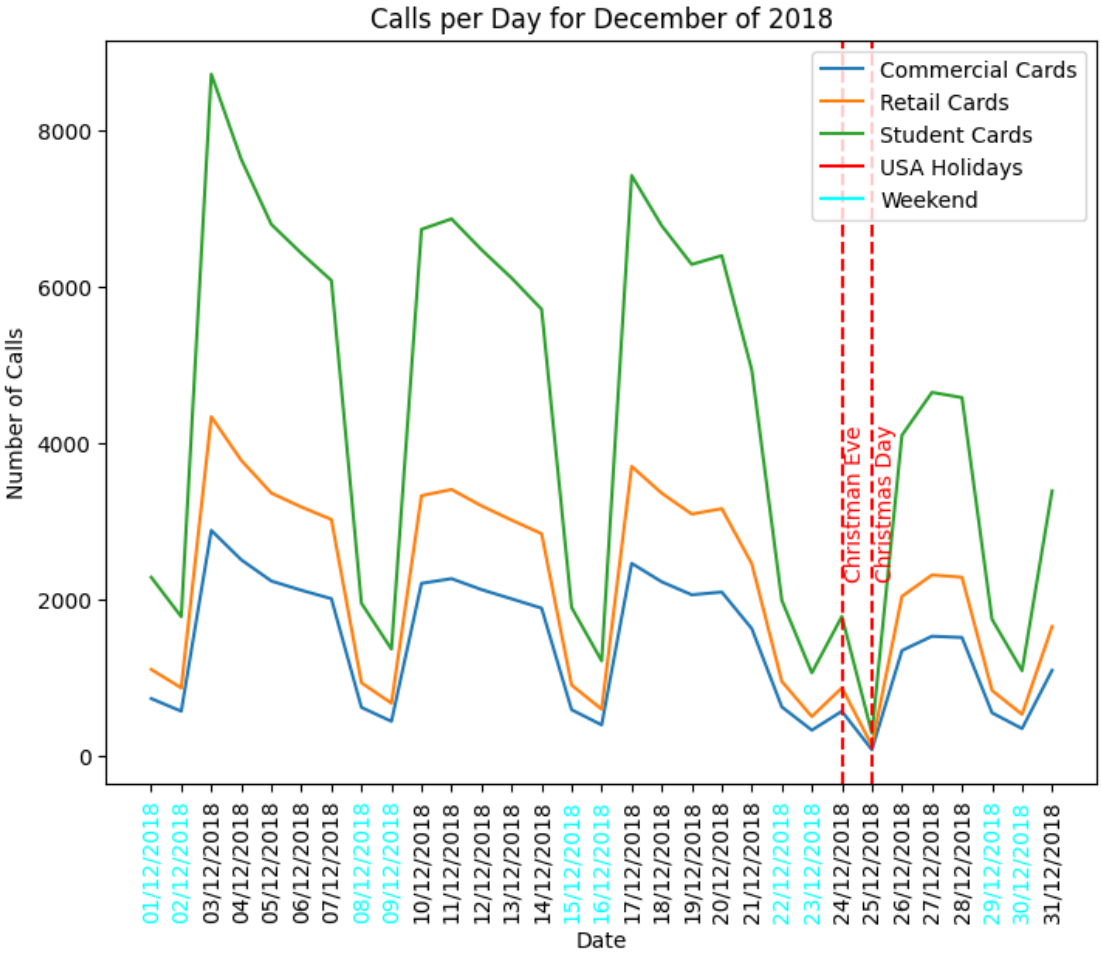

Calls per Day for July of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Independence Day
Number of Calls
0
2000
4000
6000
8000
01/07/2018
02/07/2018
03/07/2018
04/07/2018
05/07/2018
06/07/2018
07/07/2018
08/07/2018
09/07/2018
10/07/2018
11/07/2018
12/07/2018
13/07/2018
14/07/2018
15/07/2018
16/07/2018
17/07/2018
18/07/2018
19/07/2018
20/07/2018
21/07/2018
22/07/2018
23/07/2018
24/07/2018
25/07/2018
26/07/2018
27/07/2018
28/07/2018
29/07/2018
30/07/2018
31/07/2018
Date

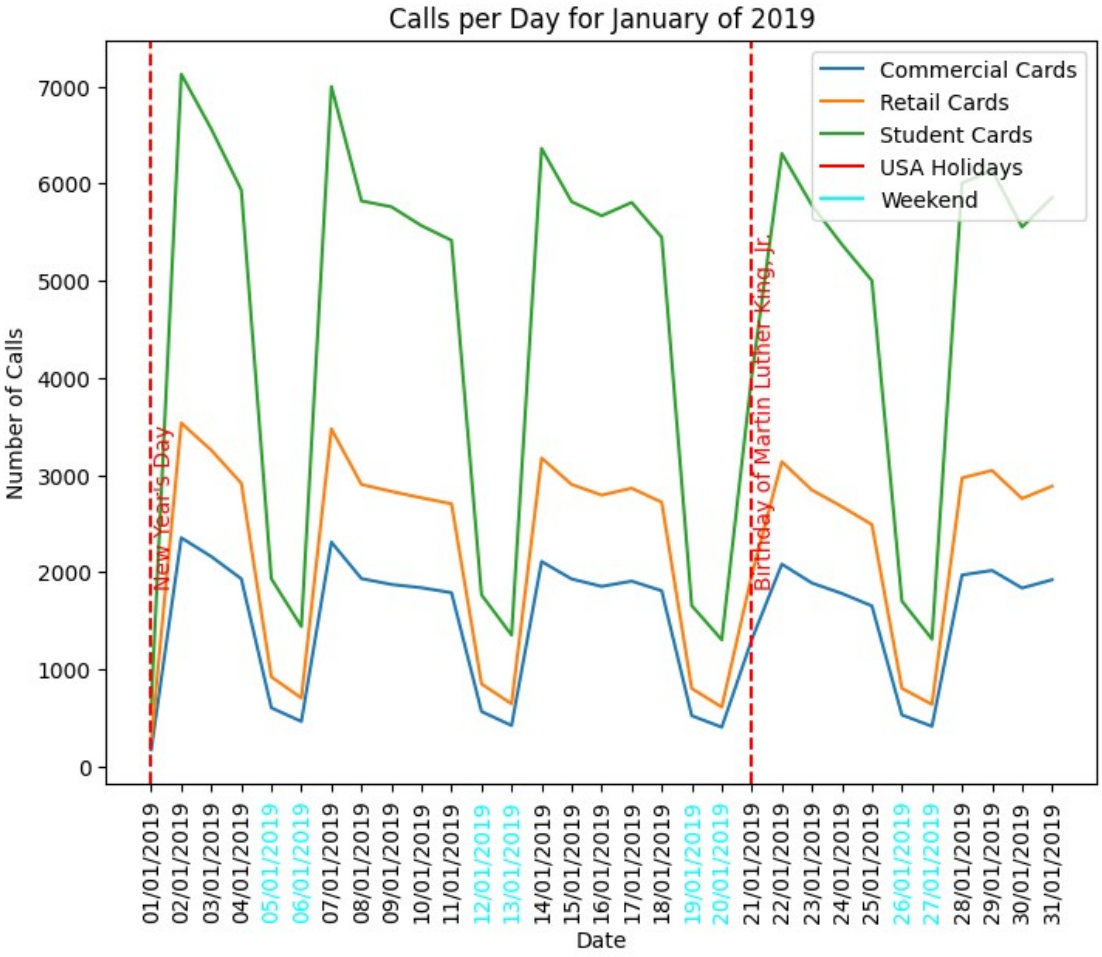

Calls per Day for August of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
2000
4000
6000
8000
01/08/2018
02/08/2018
03/08/2018
04/08/2018
05/08/2018
06/08/2018
07/08/2018
08/08/2018
09/08/2018
10/08/2018
11/08/2018
12/08/2018
13/08/2018
14/08/2018
15/08/2018
16/08/2018
17/08/2018
18/08/2018
19/08/2018
20/08/2018
21/08/2018
22/08/2018
23/08/2018
24/08/2018
25/08/2018
26/08/2018
27/08/2018
28/08/2018
29/08/2018
30/08/2018
31/08/2018
Date

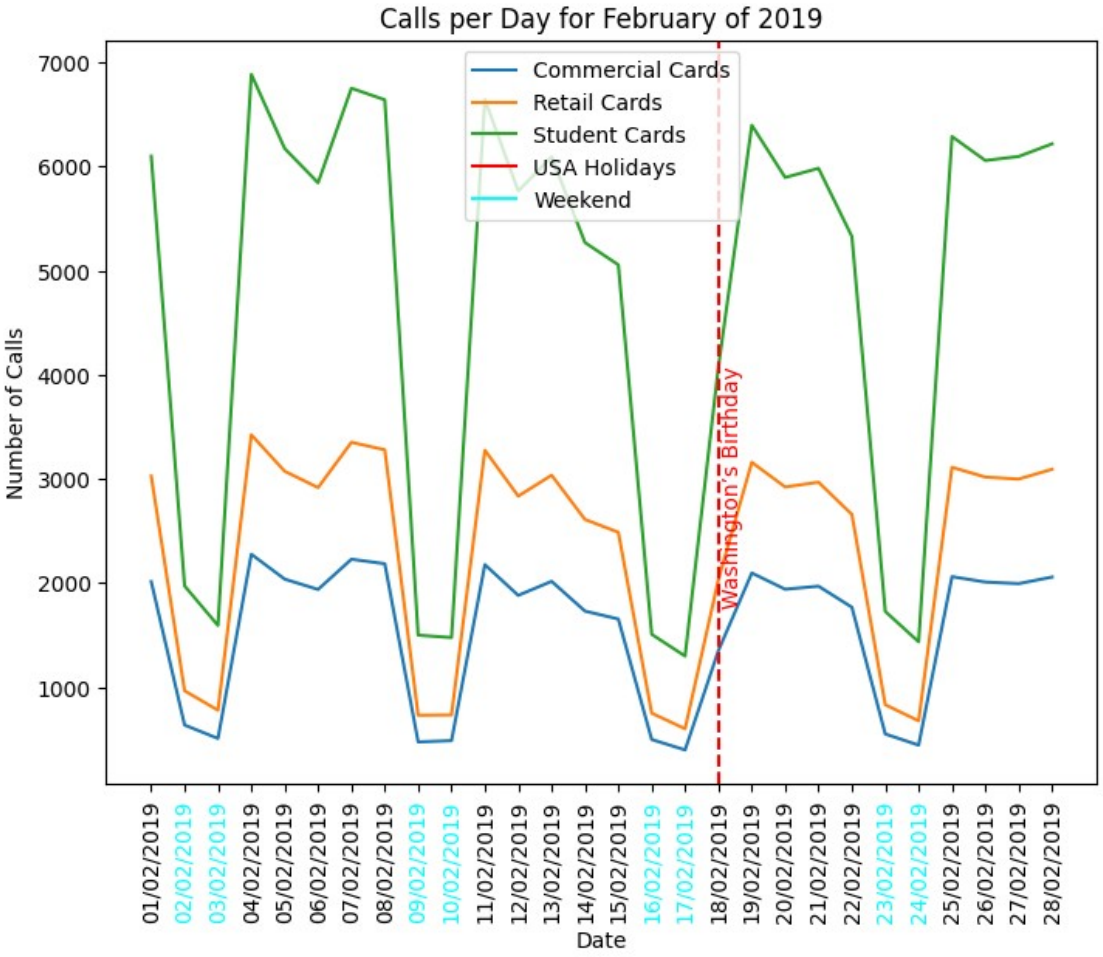

Calls per Day for September of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
2000
4000
6000
8000
Labor Day
01/09/2018
02/09/2018
03/09/2018
04/09/2018
05/09/2018
06/09/2018
07/09/2018
08/09/2018
09/09/2018
10/09/2018
11/09/2018
12/09/2018
13/09/2018
14/09/2018
15/09/2018
16/09/2018
17/09/2018
18/09/2018
19/09/2018
20/09/2018
21/09/2018
22/09/2018
23/09/2018
24/09/2018
25/09/2018
26/09/2018
27/09/2018
28/09/2018
29/09/2018
30/09/2018
Date

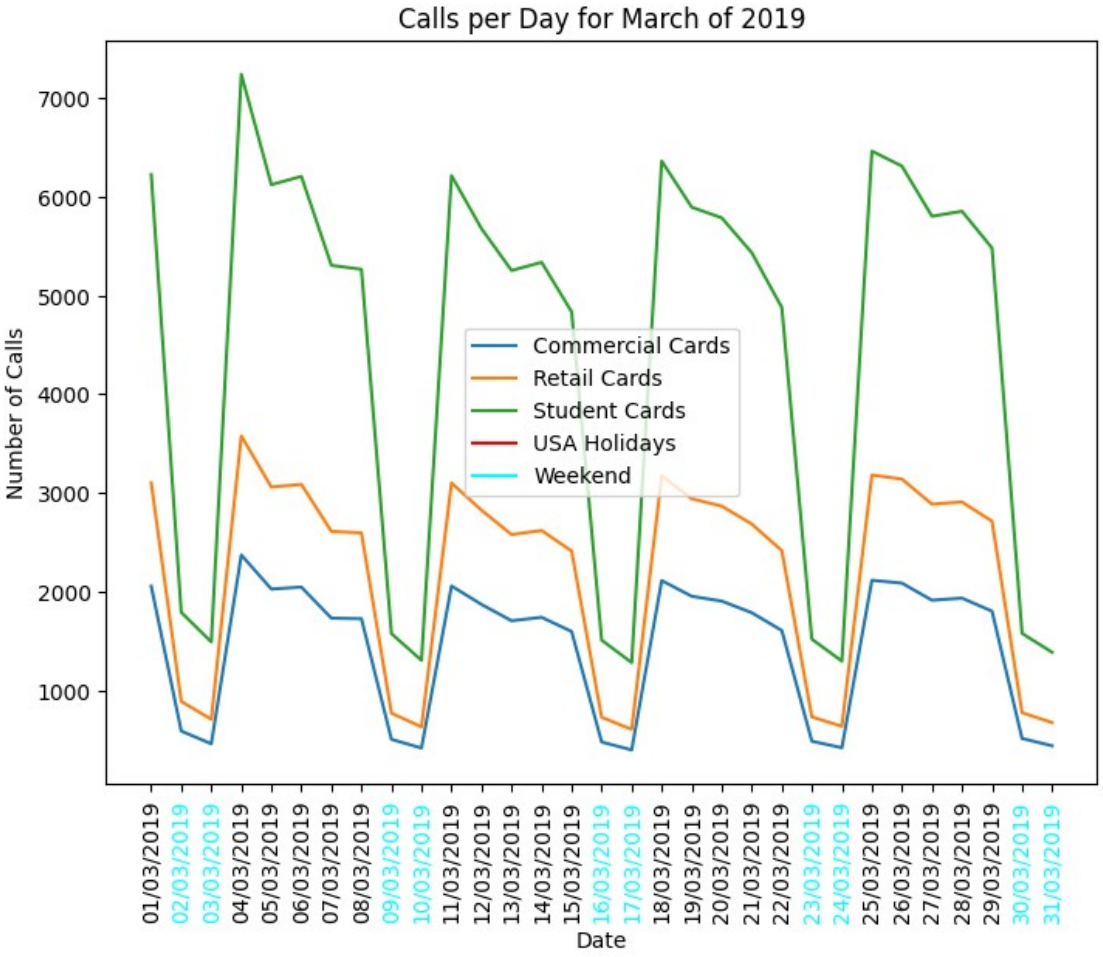

Calls per Day for October of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Columbus Day
Number of Calls
8000
7000
6000
5000
4000
3000
2000
1000
01/10/2018
02/10/2018
03/10/2018
04/10/2018
05/10/2018
06/10/2018
07/10/2018
08/10/2018
09/10/2018
10/10/2018
11/10/2018
12/10/2018
13/10/2018
14/10/2018
15/10/2018
16/10/2018
17/10/2018
18/10/2018
19/10/2018
20/10/2018
21/10/2018
22/10/2018
23/10/2018
24/10/2018
25/10/2018
26/10/2018
27/10/2018
28/10/2018
29/10/2018
30/10/2018
31/10/2018
Date

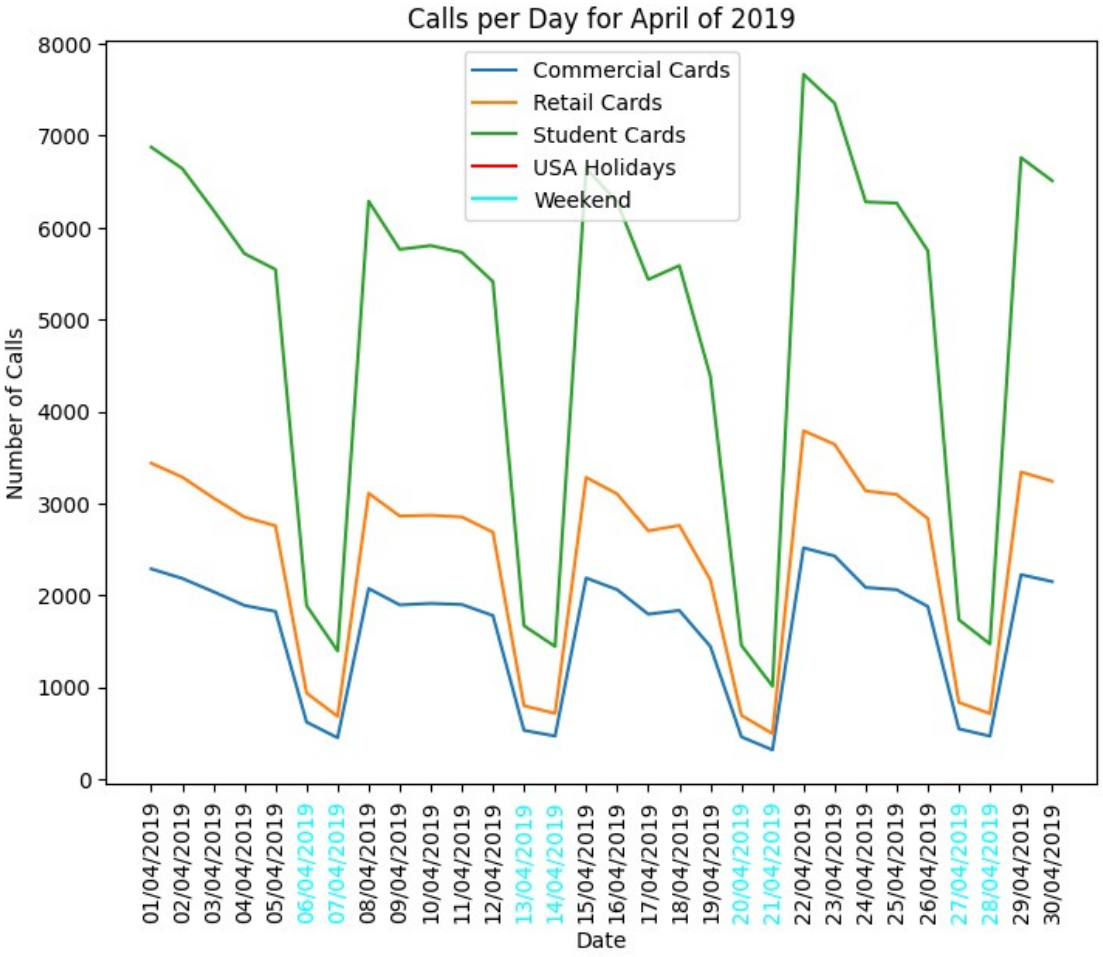

Calls per Day for November of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
2000
4000
6000
8000
Veterans Day
Thanksgiving Day
Black Friday
01/11/2018
02/11/2018
03/11/2018
04/11/2018
05/11/2018
06/11/2018
07/11/2018
08/11/2018
09/11/2018
10/11/2018
11/11/2018
12/11/2018
13/11/2018
14/11/2018
15/11/2018
16/11/2018
17/11/2018
18/11/2018
19/11/2018
20/11/2018
21/11/2018
22/11/2018
23/11/2018
24/11/2018
25/11/2018
26/11/2018
27/11/2018
28/11/2018
29/11/2018
30/11/2018
Date

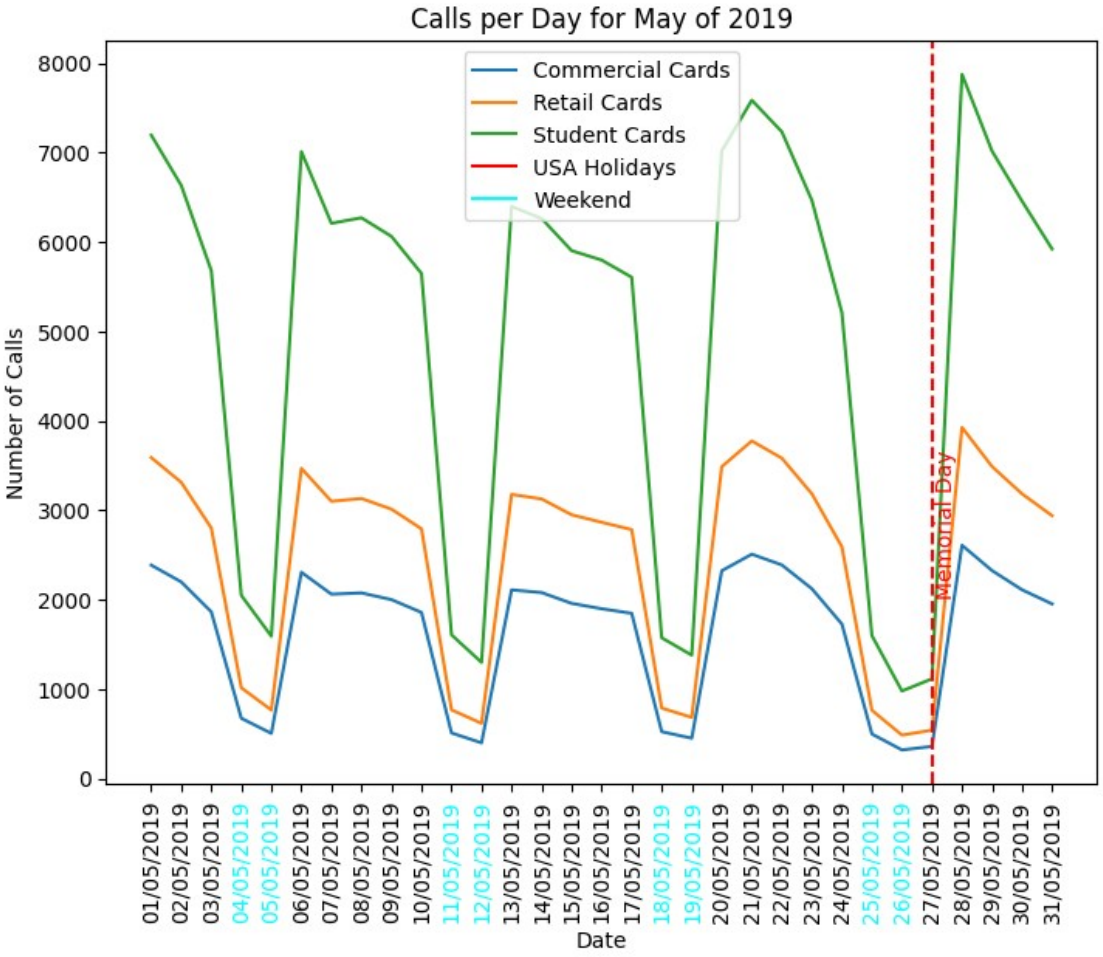

Calls per Day for December of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
2000
4000
6000
8000
Christman Eve
Christmas Day
01/12/2018
02/12/2018
03/12/2018
04/12/2018
05/12/2018
06/12/2018
07/12/2018
08/12/2018
09/12/2018
10/12/2018
11/12/2018
12/12/2018
13/12/2018
14/12/2018
15/12/2018
16/12/2018
17/12/2018
18/12/2018
19/12/2018
20/12/2018
21/12/2018
22/12/2018
23/12/2018
24/12/2018
25/12/2018
26/12/2018
27/12/2018
28/12/2018
29/12/2018
30/12/2018
31/12/2018
Date

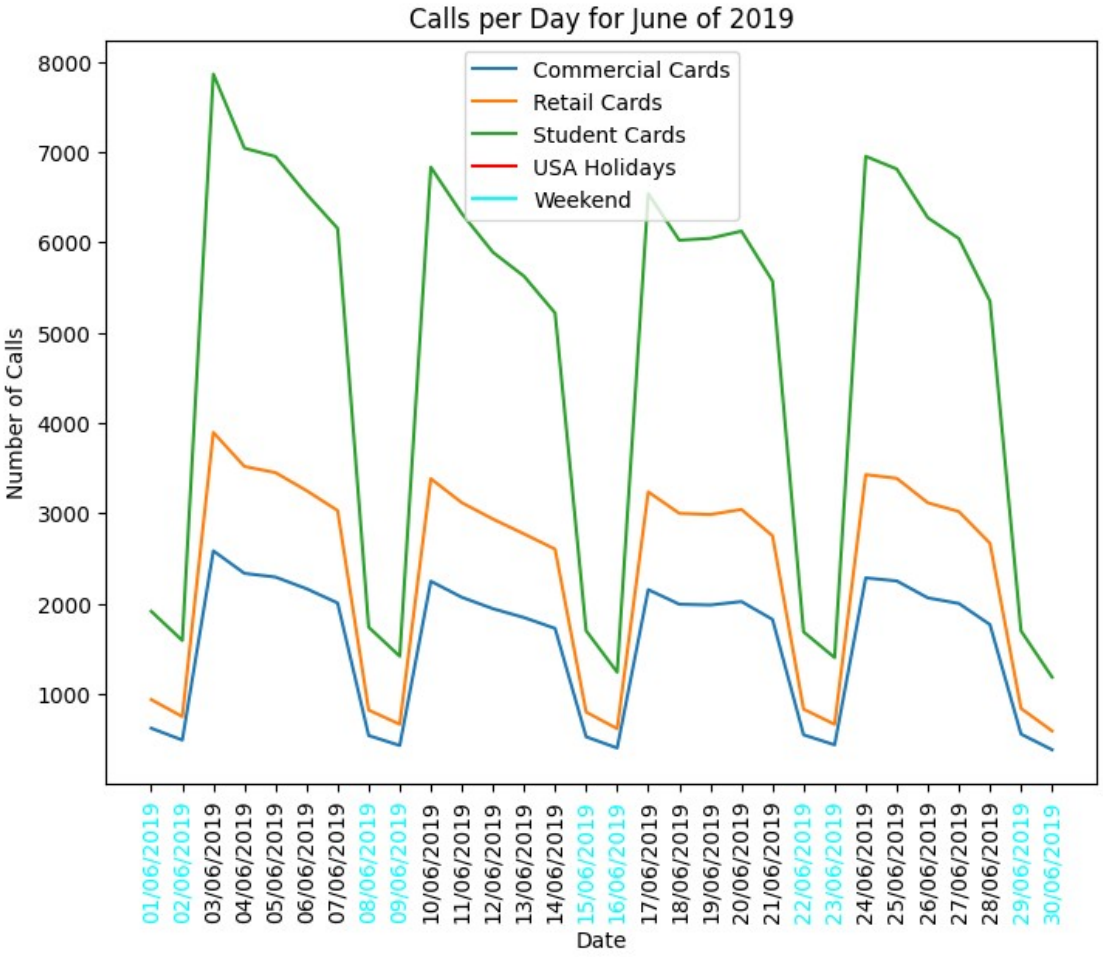

Calls per Day for January of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
New Year's Day
Birthday of Martin Luther King, Jr.
01/01/2019
02/01/2019
03/01/2019
04/01/2019
05/01/2019
06/01/2019
07/01/2019
08/01/2019
09/01/2019
10/01/2019
11/01/2019
12/01/2019
13/01/2019
14/01/2019
15/01/2019
16/01/2019
17/01/2019
18/01/2019
19/01/2019
20/01/2019
21/01/2019
22/01/2019
23/01/2019
24/01/2019
25/01/2019
26/01/2019
27/01/2019
28/01/2019
29/01/2019
30/01/2019
31/01/2019
Date

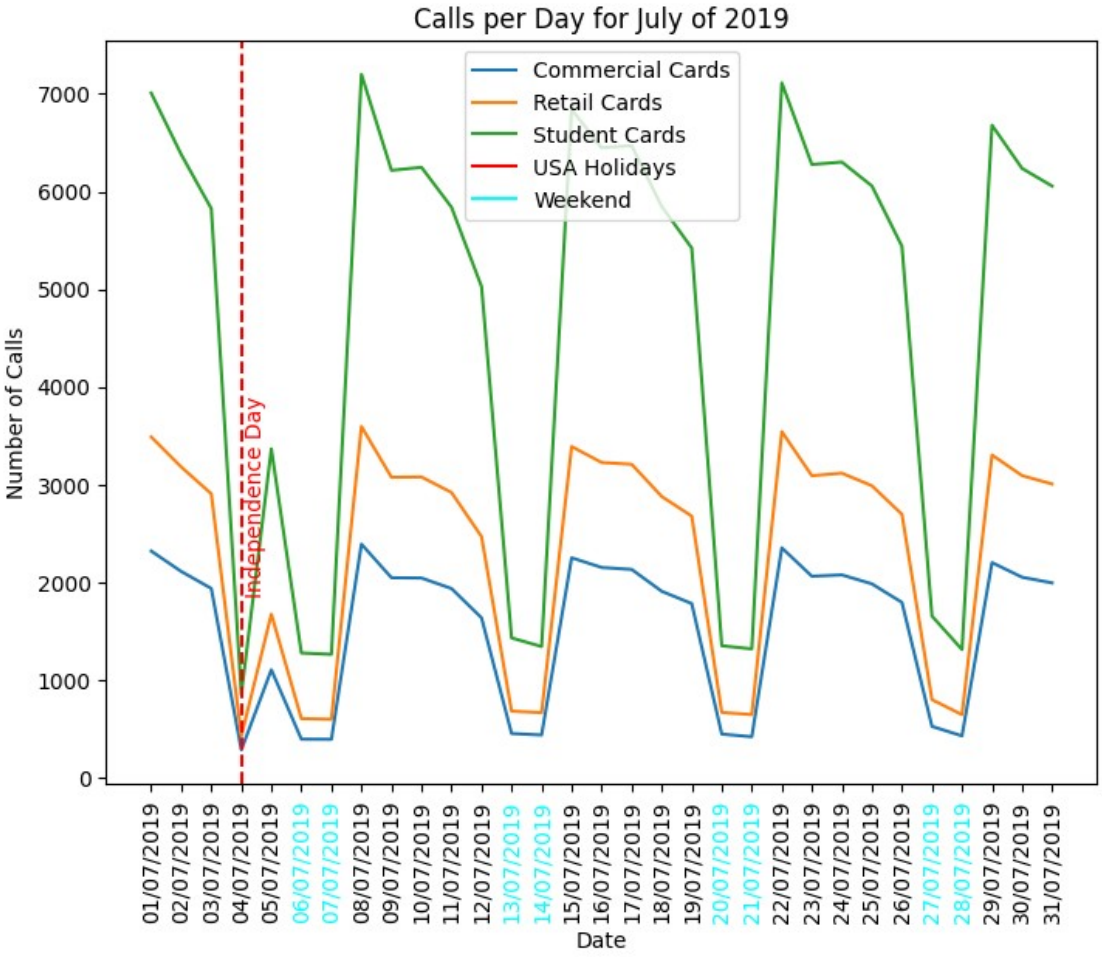

Calls per Day for February of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Washington's Birthday
Number of Calls
7000
6000
5000
4000
3000
2000
1000
01/02/2019
02/02/2019
03/02/2019
04/02/2019
05/02/2019
06/02/2019
07/02/2019
08/02/2019
09/02/2019
10/02/2019
11/02/2019
12/02/2019
13/02/2019
14/02/2019
15/02/2019
16/02/2019
17/02/2019
18/02/2019
19/02/2019
20/02/2019
21/02/2019
22/02/2019
23/02/2019
24/02/2019
25/02/2019
26/02/2019
27/02/2019
28/02/2019
Date

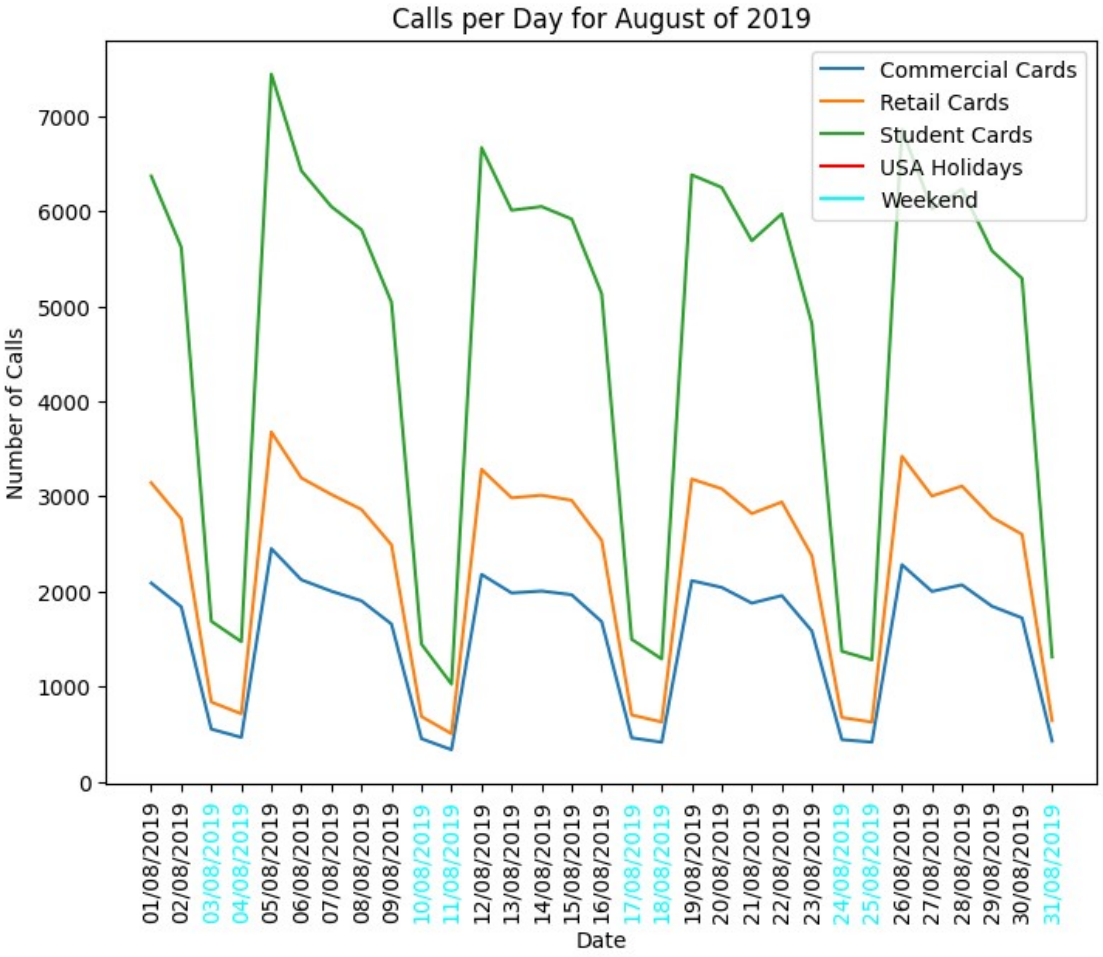

Calls per Day for March of 2019
Number of Calls
7000
6000
5000
4000
3000
2000
1000
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
01/03/2019
02/03/2019
03/03/2019
04/03/2019
05/03/2019
06/03/2019
07/03/2019
08/03/2019
09/03/2019
10/03/2019
11/03/2019
12/03/2019
13/03/2019
14/03/2019
15/03/2019
16/03/2019
17/03/2019
18/03/2019
19/03/2019
20/03/2019
21/03/2019
22/03/2019
23/03/2019
24/03/2019
25/03/2019
26/03/2019
27/03/2019
28/03/2019
29/03/2019
30/03/2019
31/03/2019
Date

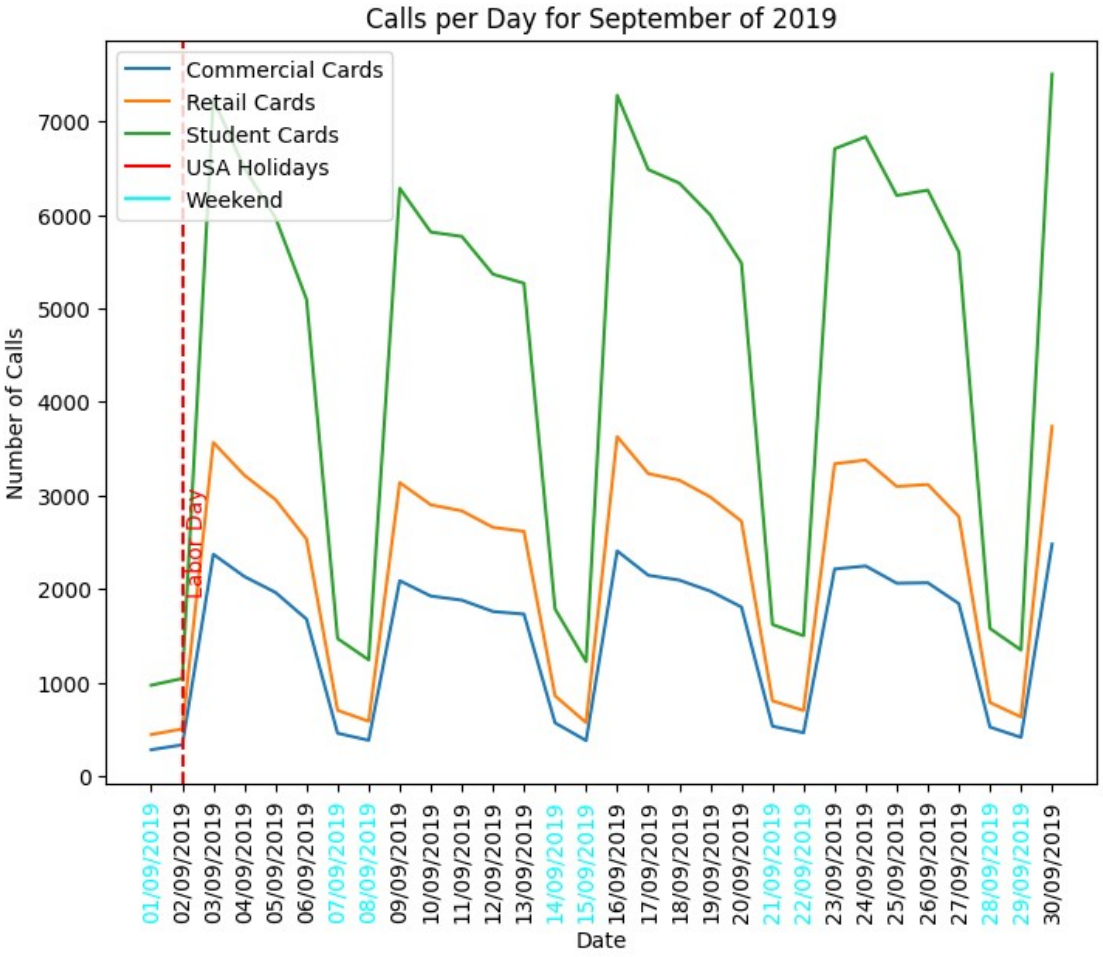

Calls per Day for April of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
8000
01/04/2019
02/04/2019
03/04/2019
04/04/2019
05/04/2019
06/04/2019
07/04/2019
08/04/2019
09/04/2019
10/04/2019
11/04/2019
12/04/2019
13/04/2019
14/04/2019
15/04/2019
16/04/2019
17/04/2019
18/04/2019
19/04/2019
20/04/2019
21/04/2019
22/04/2019
23/04/2019
24/04/2019
25/04/2019
26/04/2019
27/04/2019
28/04/2019
29/04/2019
30/04/2019
Date

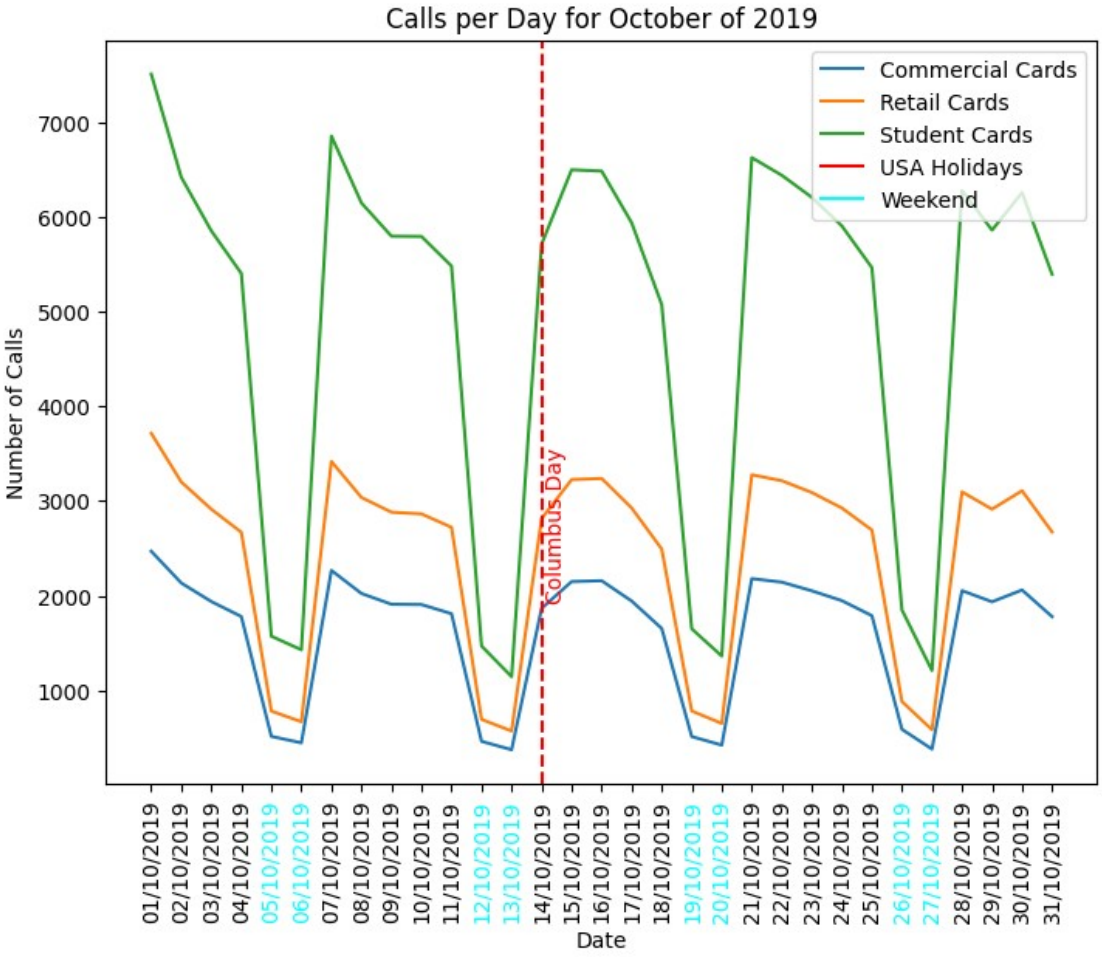

Calls per Day for May of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Memorial Day
Number of Calls
Date
0
1000
2000
3000
4000
5000
6000
7000
8000
01/05/2019
02/05/2019
03/05/2019
04/05/2019
05/05/2019
06/05/2019
07/05/2019
08/05/2019
09/05/2019
10/05/2019
11/05/2019
12/05/2019
13/05/2019
14/05/2019
15/05/2019
16/05/2019
17/05/2019
18/05/2019
19/05/2019
20/05/2019
21/05/2019
22/05/2019
23/05/2019
24/05/2019
25/05/2019
26/05/2019
27/05/2019
28/05/2019
29/05/2019
30/05/2019
31/05/2019

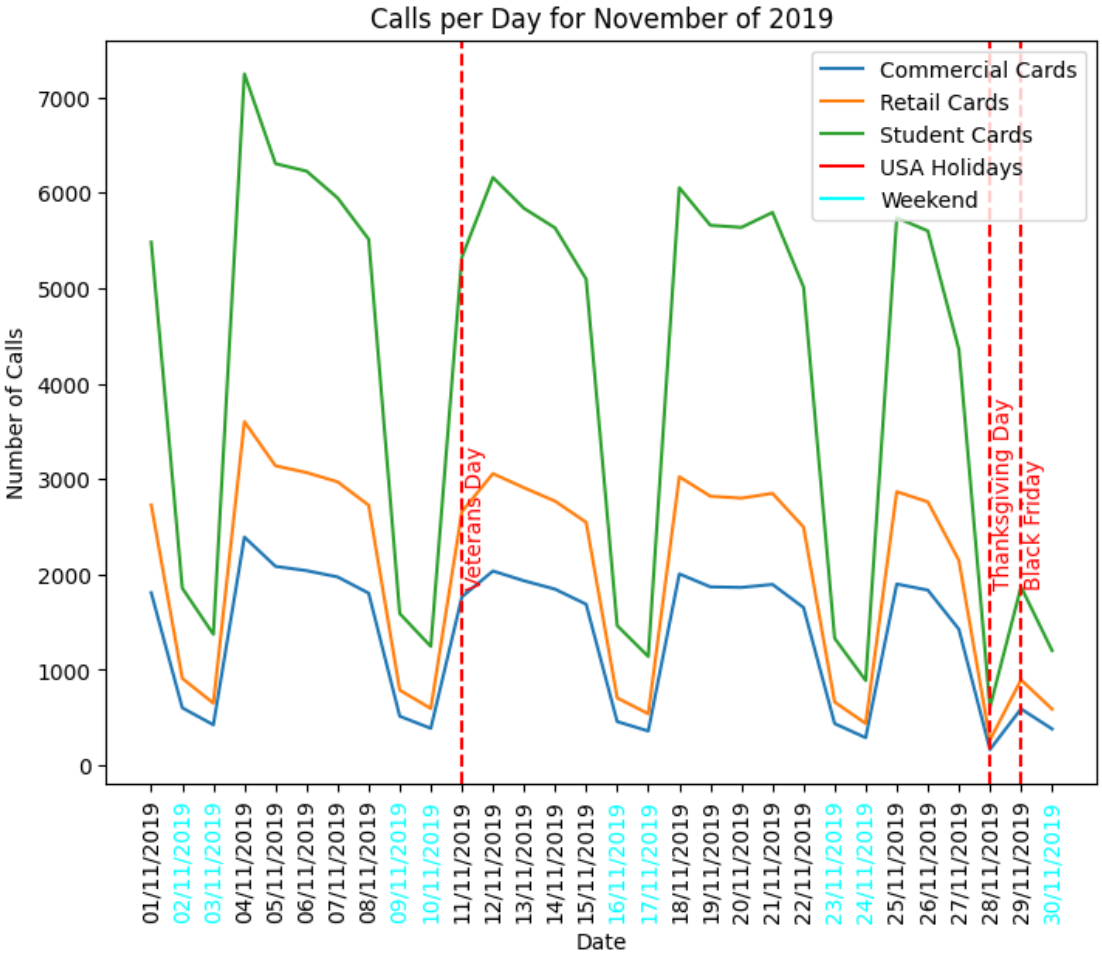

Calls per Day for June of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
8000
7000
6000
5000
4000
3000
2000
1000
01/06/2019
02/06/2019
03/06/2019
04/06/2019
05/06/2019
06/06/2019
07/06/2019
08/06/2019
09/06/2019
10/06/2019
11/06/2019
12/06/2019
13/06/2019
14/06/2019
15/06/2019
16/06/2019
17/06/2019
18/06/2019
19/06/2019
20/06/2019
21/06/2019
22/06/2019
23/06/2019
24/06/2019
25/06/2019
26/06/2019
27/06/2019
28/06/2019
29/06/2019
30/06/2019
Date

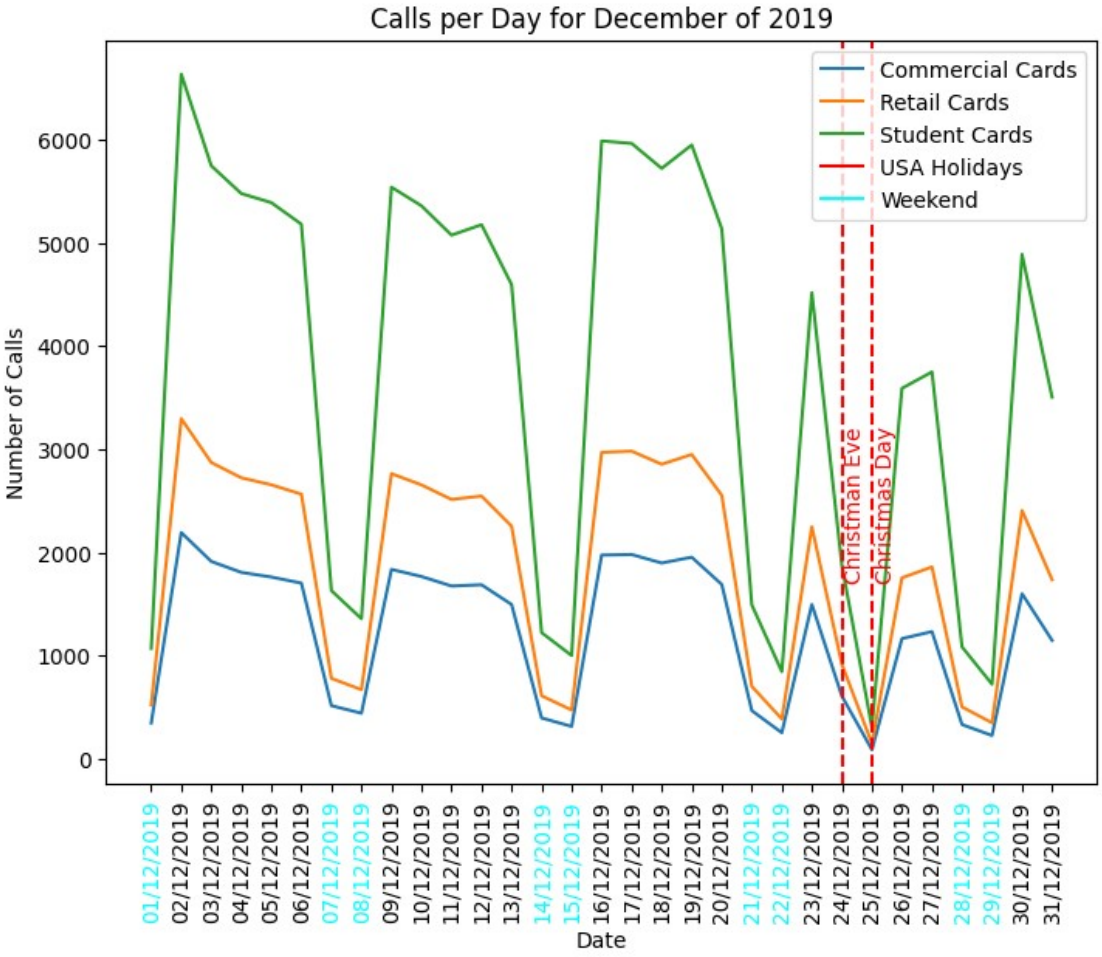

Calls per Day for July of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Independence Day
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/07/2019
02/07/2019
03/07/2019
04/07/2019
05/07/2019
06/07/2019
07/07/2019
08/07/2019
09/07/2019
10/07/2019
11/07/2019
12/07/2019
13/07/2019
14/07/2019
15/07/2019
16/07/2019
17/07/2019
18/07/2019
19/07/2019
20/07/2019
21/07/2019
22/07/2019
23/07/2019
24/07/2019
25/07/2019
26/07/2019
27/07/2019
28/07/2019
29/07/2019
30/07/2019
31/07/2019
Date

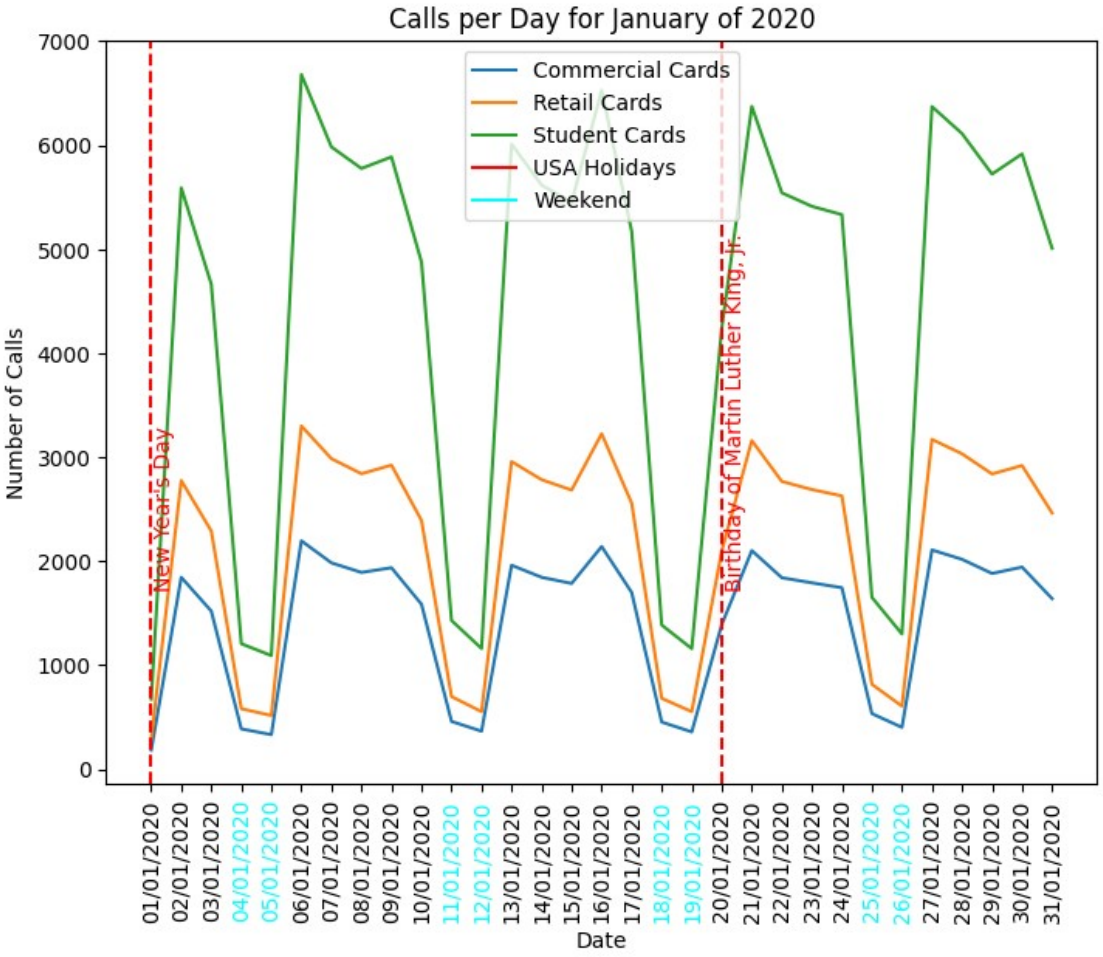

Calls per Day for August of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
Date
01/08/2019
02/08/2019
03/08/2019
04/08/2019
05/08/2019
06/08/2019
07/08/2019
08/08/2019
09/08/2019
10/08/2019
11/08/2019
12/08/2019
13/08/2019
14/08/2019
15/08/2019
16/08/2019
17/08/2019
18/08/2019
19/08/2019
20/08/2019
21/08/2019
22/08/2019
23/08/2019
24/08/2019
25/08/2019
26/08/2019
27/08/2019
28/08/2019
29/08/2019
30/08/2019
31/08/2019

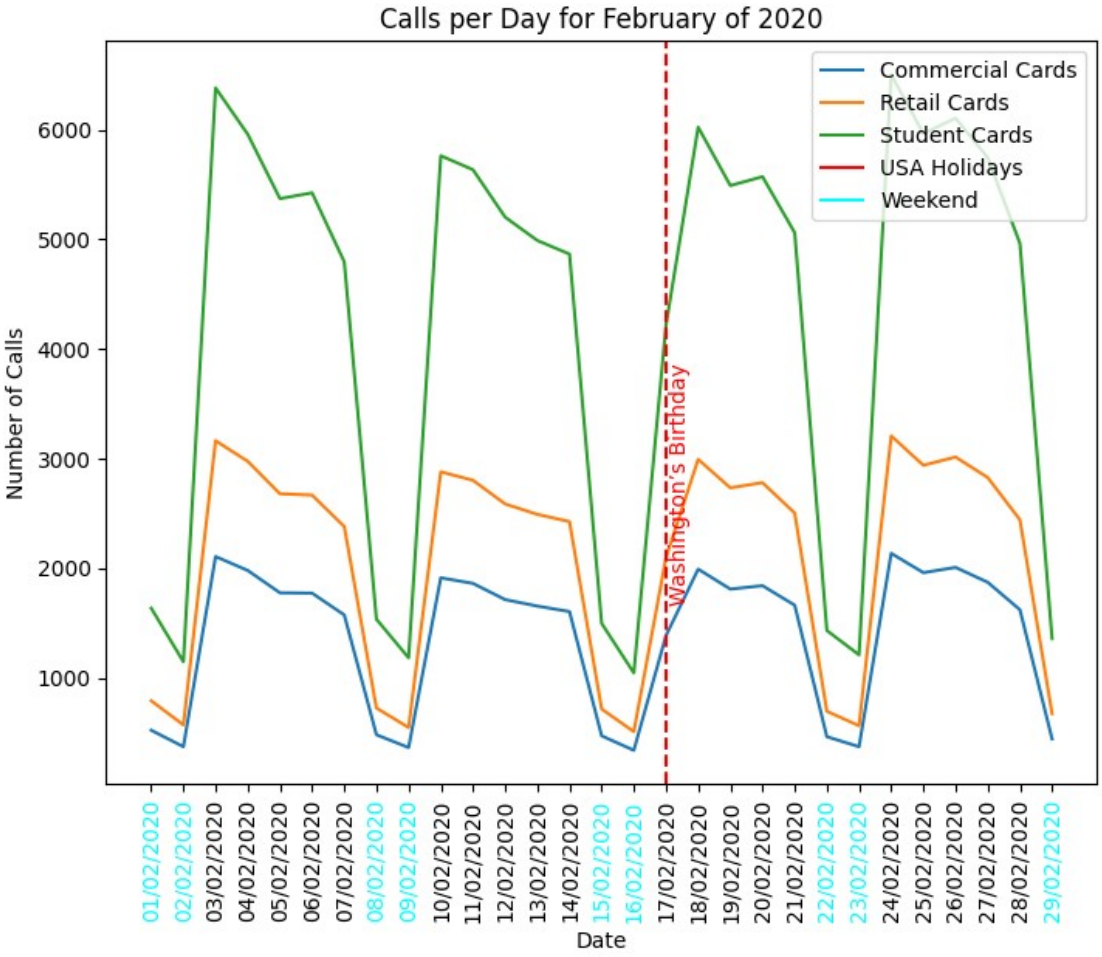

Calls per Day for September of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
Labor Day
01/09/2019
02/09/2019
03/09/2019
04/09/2019
05/09/2019
06/09/2019
07/09/2019
08/09/2019
09/09/2019
10/09/2019
11/09/2019
12/09/2019
13/09/2019
14/09/2019
15/09/2019
16/09/2019
17/09/2019
18/09/2019
19/09/2019
20/09/2019
21/09/2019
22/09/2019
23/09/2019
24/09/2019
25/09/2019
26/09/2019
27/09/2019
28/09/2019
29/09/2019
30/09/2019
Date

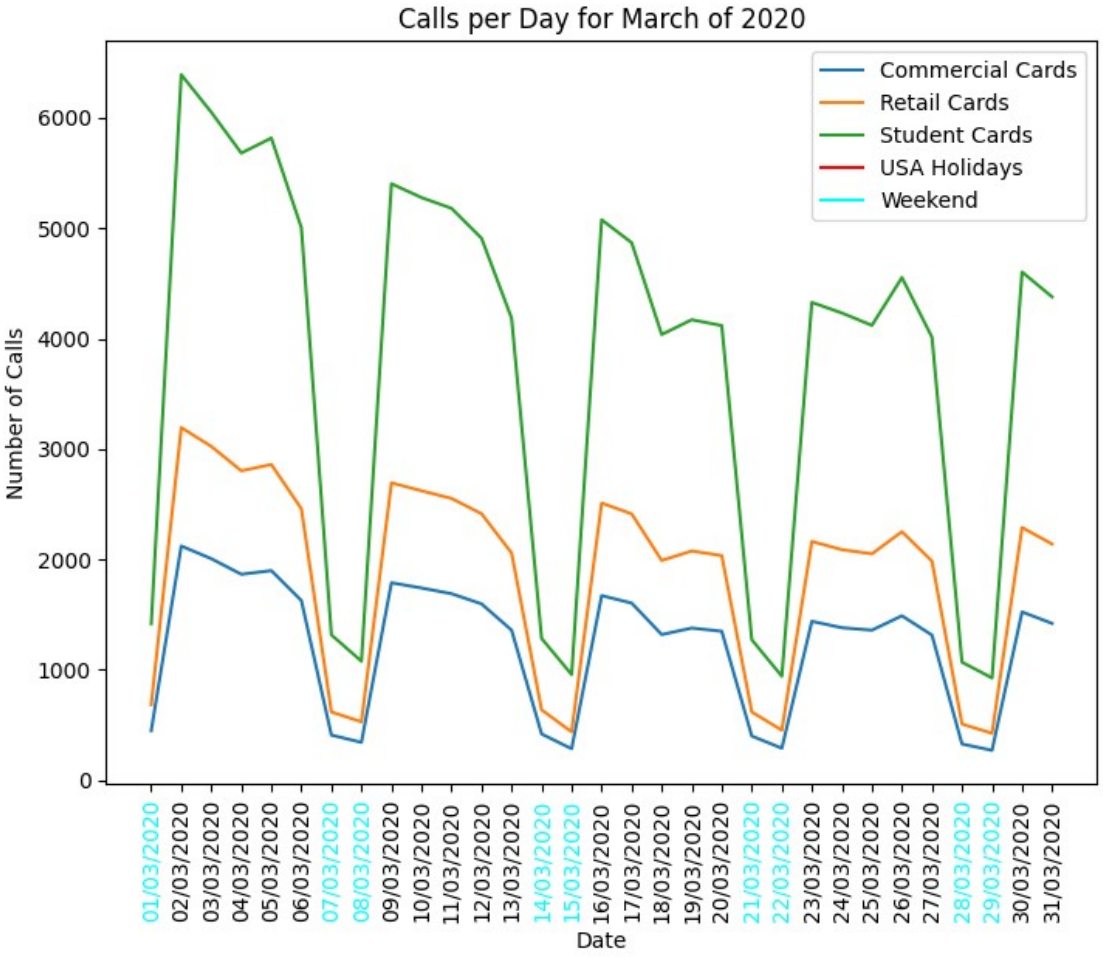

Calls per Day for October of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Columbus Day
Number of Calls
7000
6000
5000
4000
3000
2000
1000
01/10/2019
02/10/2019
03/10/2019
04/10/2019
05/10/2019
06/10/2019
07/10/2019
08/10/2019
09/10/2019
10/10/2019
11/10/2019
12/10/2019
13/10/2019
14/10/2019
15/10/2019
16/10/2019
17/10/2019
18/10/2019
19/10/2019
20/10/2019
21/10/2019
22/10/2019
23/10/2019
24/10/2019
25/10/2019
26/10/2019
27/10/2019
28/10/2019
29/10/2019
30/10/2019
31/10/2019
Date

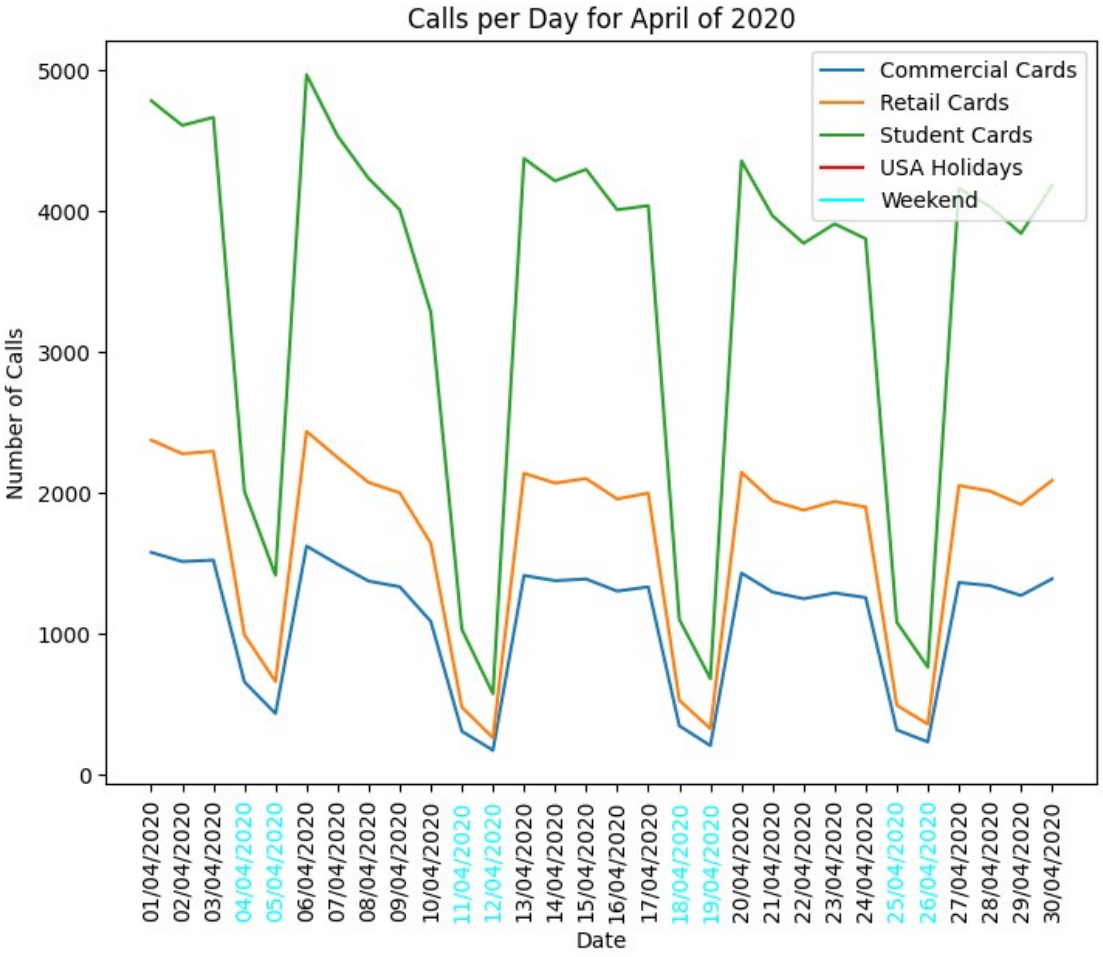

Calls per Day for November of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
Veterans Day
Thanksgiving Day
Black Friday
01/11/2019
02/11/2019
03/11/2019
04/11/2019
05/11/2019
06/11/2019
07/11/2019
08/11/2019
09/11/2019
10/11/2019
11/11/2019
12/11/2019
13/11/2019
14/11/2019
15/11/2019
16/11/2019
17/11/2019
18/11/2019
19/11/2019
20/11/2019
21/11/2019
22/11/2019
23/11/2019
24/11/2019
25/11/2019
26/11/2019
27/11/2019
28/11/2019
29/11/2019
30/11/2019
Date

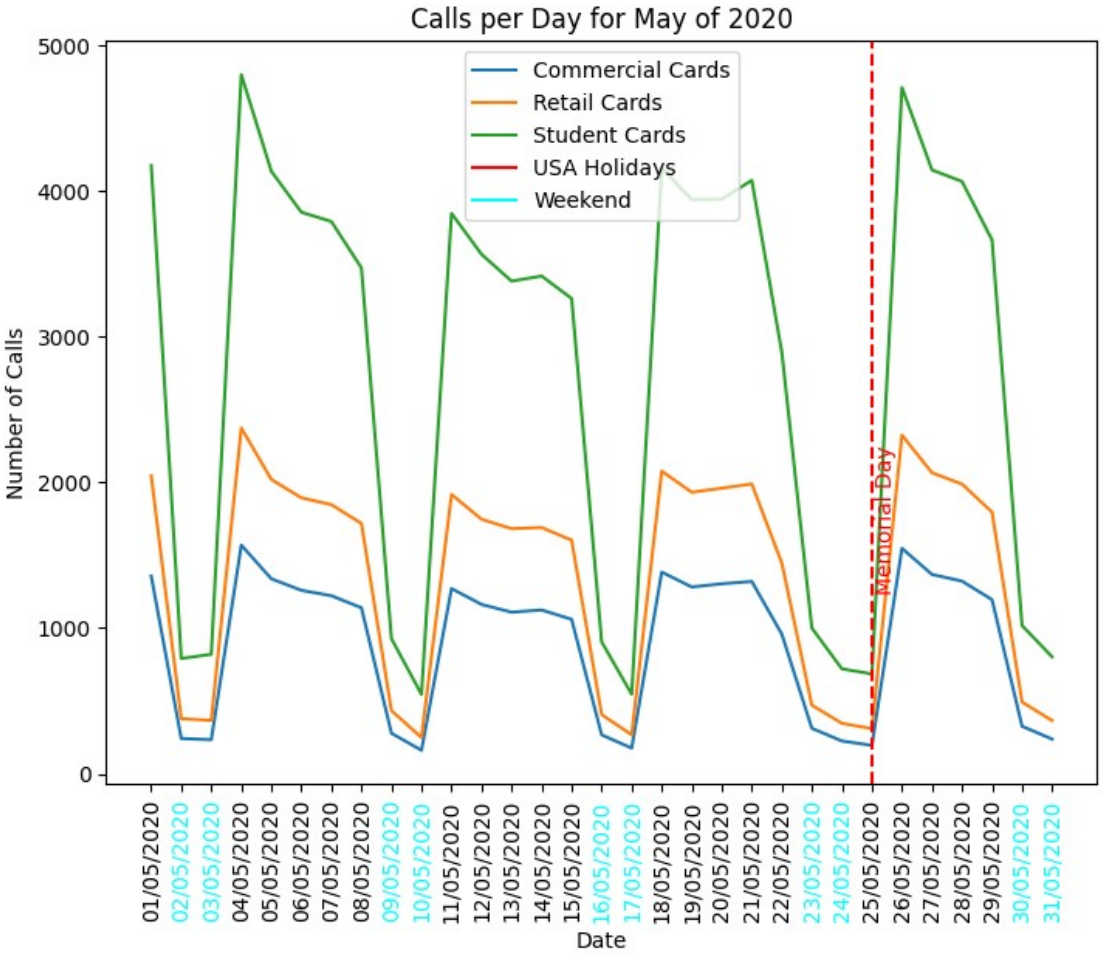

Calls per Day for December of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
Christman Eve
Christmas Day
01/12/2019
02/12/2019
03/12/2019
04/12/2019
05/12/2019
06/12/2019
07/12/2019
08/12/2019
09/12/2019
10/12/2019
11/12/2019
12/12/2019
13/12/2019
14/12/2019
15/12/2019
16/12/2019
17/12/2019
18/12/2019
19/12/2019
20/12/2019
21/12/2019
22/12/2019
23/12/2019
24/12/2019
25/12/2019
26/12/2019
27/12/2019
28/12/2019
29/12/2019
30/12/2019
31/12/2019
Date

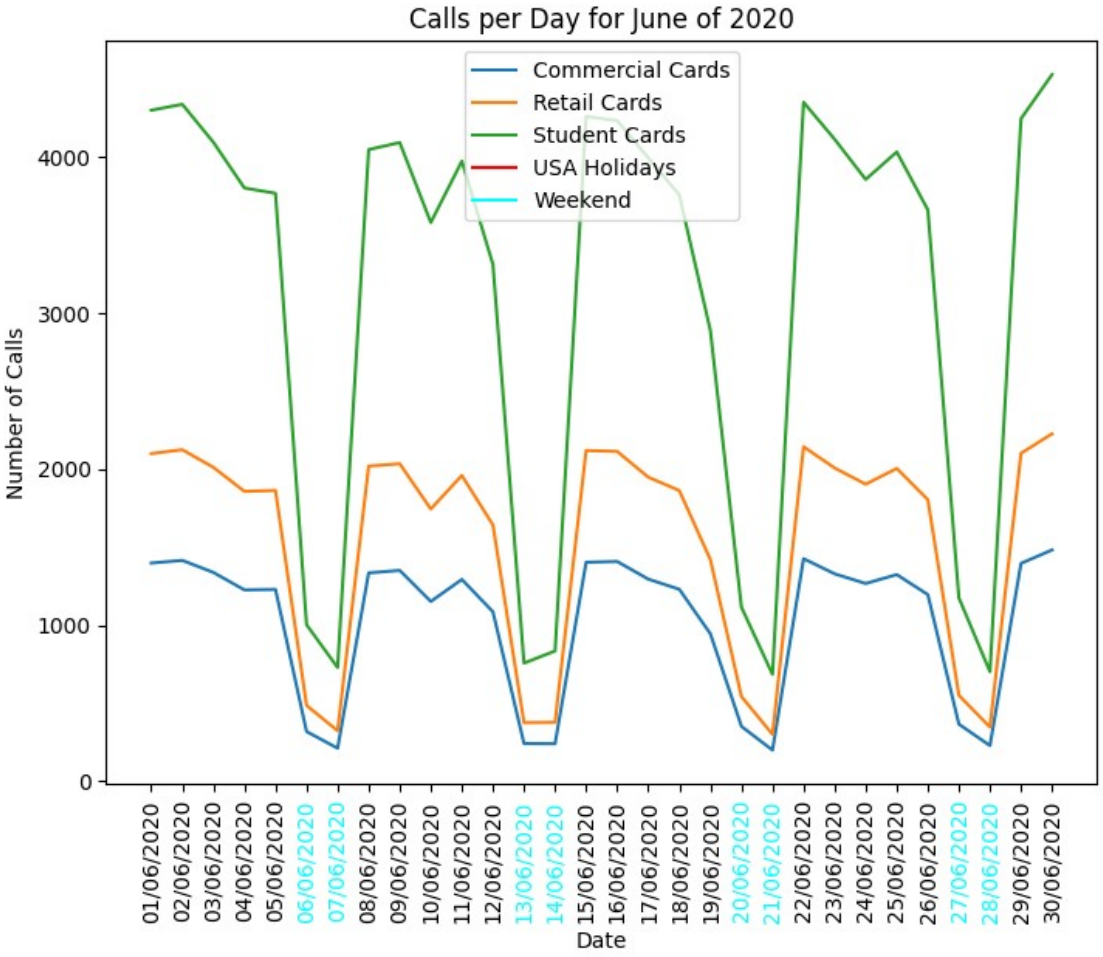

Calls per Day for January of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
New Year's Day
Birthday of Martin Luther King, Jr.
Number of Calls
Date
0
1000
2000
3000
4000
5000
6000
7000
01/01/2020
02/01/2020
03/01/2020
04/01/2020
05/01/2020
06/01/2020
07/01/2020
08/01/2020
09/01/2020
10/01/2020
11/01/2020
12/01/2020
13/01/2020
14/01/2020
15/01/2020
16/01/2020
17/01/2020
18/01/2020
19/01/2020
20/01/2020
21/01/2020
22/01/2020
23/01/2020
24/01/2020
25/01/2020
26/01/2020
27/01/2020
28/01/2020
29/01/2020
30/01/2020
31/01/2020

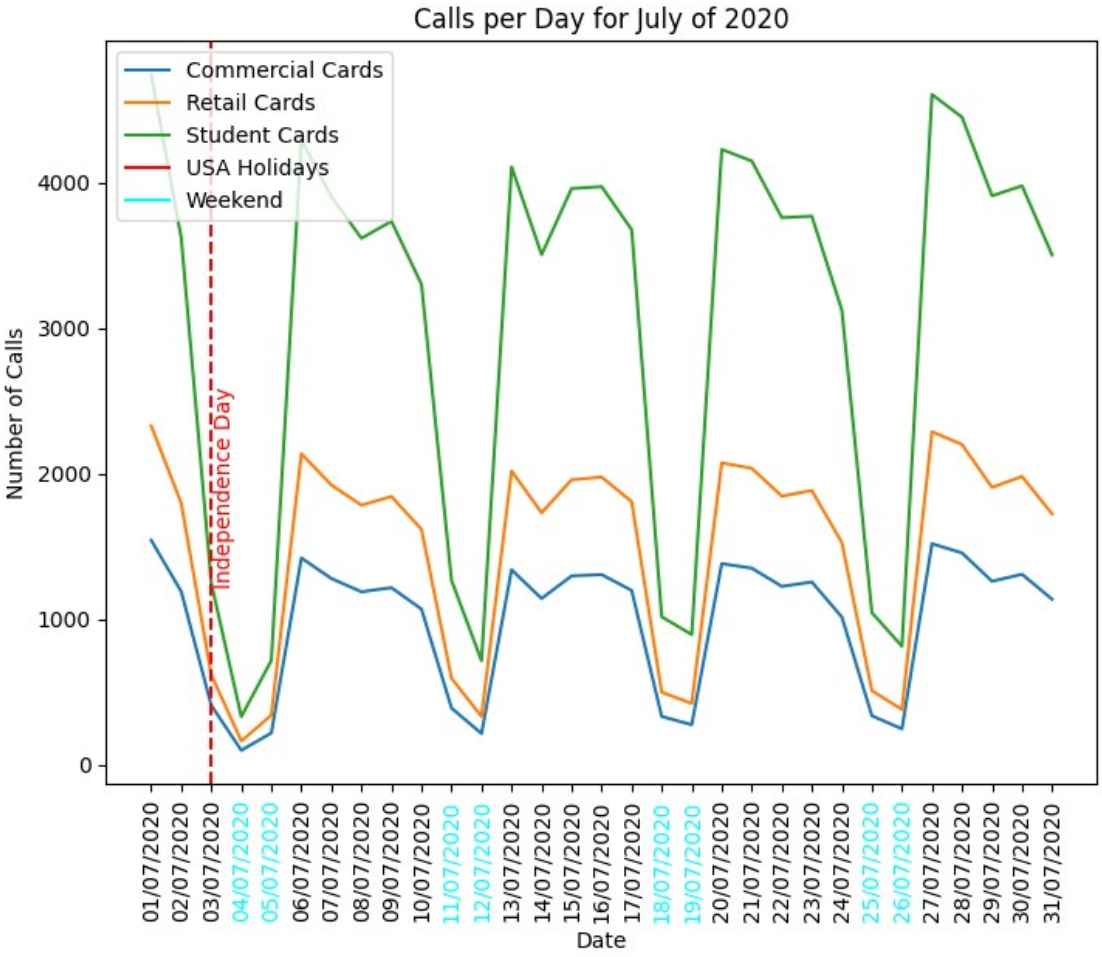

Calls per Day for February of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
6000
5000
4000
3000
2000
1000
Washington's Birthday
01/02/2020
02/02/2020
03/02/2020
04/02/2020
05/02/2020
06/02/2020
07/02/2020
08/02/2020
09/02/2020
10/02/2020
11/02/2020
12/02/2020
13/02/2020
14/02/2020
15/02/2020
16/02/2020
17/02/2020
18/02/2020
19/02/2020
20/02/2020
21/02/2020
22/02/2020
23/02/2020
24/02/2020
25/02/2020
26/02/2020
27/02/2020
28/02/2020
29/02/2020
Date

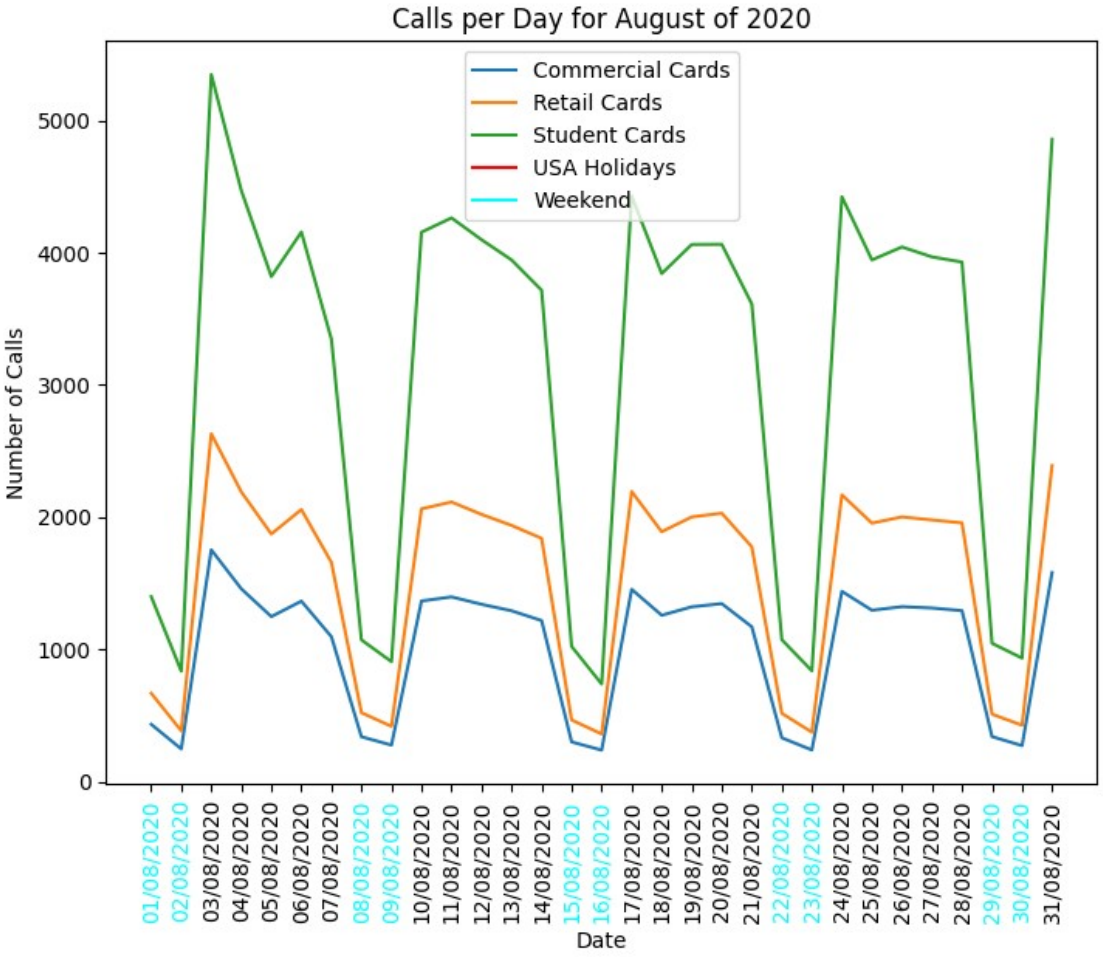

Calls per Day for March of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
Date
01/03/2020
02/03/2020
03/03/2020
04/03/2020
05/03/2020
06/03/2020
07/03/2020
08/03/2020
09/03/2020
10/03/2020
11/03/2020
12/03/2020
13/03/2020
14/03/2020
15/03/2020
16/03/2020
17/03/2020
18/03/2020
19/03/2020
20/03/2020
21/03/2020
22/03/2020
23/03/2020
24/03/2020
25/03/2020
26/03/2020
27/03/2020
28/03/2020
29/03/2020
30/03/2020
31/03/2020

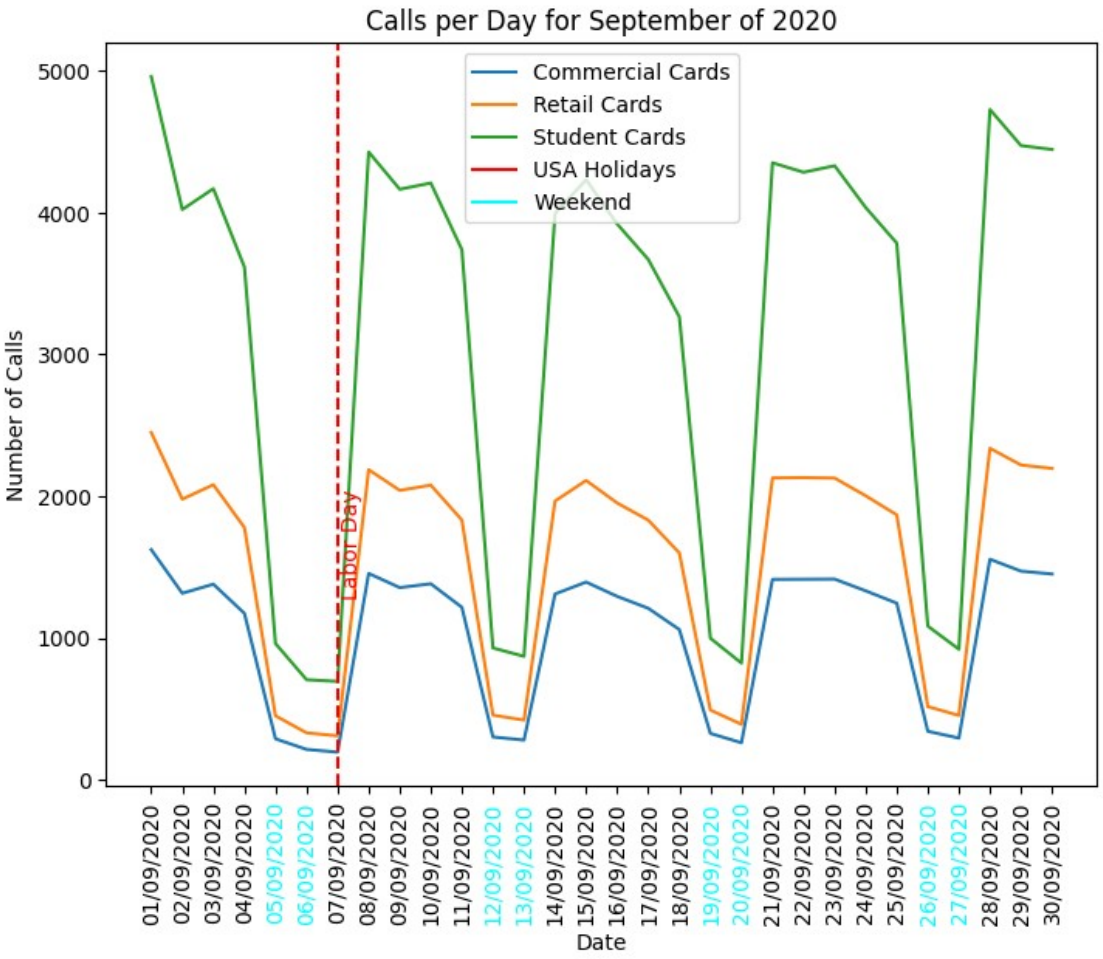

Calls per Day for April of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
01/04/2020
02/04/2020
03/04/2020
04/04/2020
05/04/2020
06/04/2020
07/04/2020
08/04/2020
09/04/2020
10/04/2020
11/04/2020
12/04/2020
13/04/2020
14/04/2020
15/04/2020
16/04/2020
17/04/2020
18/04/2020
19/04/2020
20/04/2020
21/04/2020
22/04/2020
23/04/2020
24/04/2020
25/04/2020
26/04/2020
27/04/2020
28/04/2020
29/04/2020
30/04/2020
Date

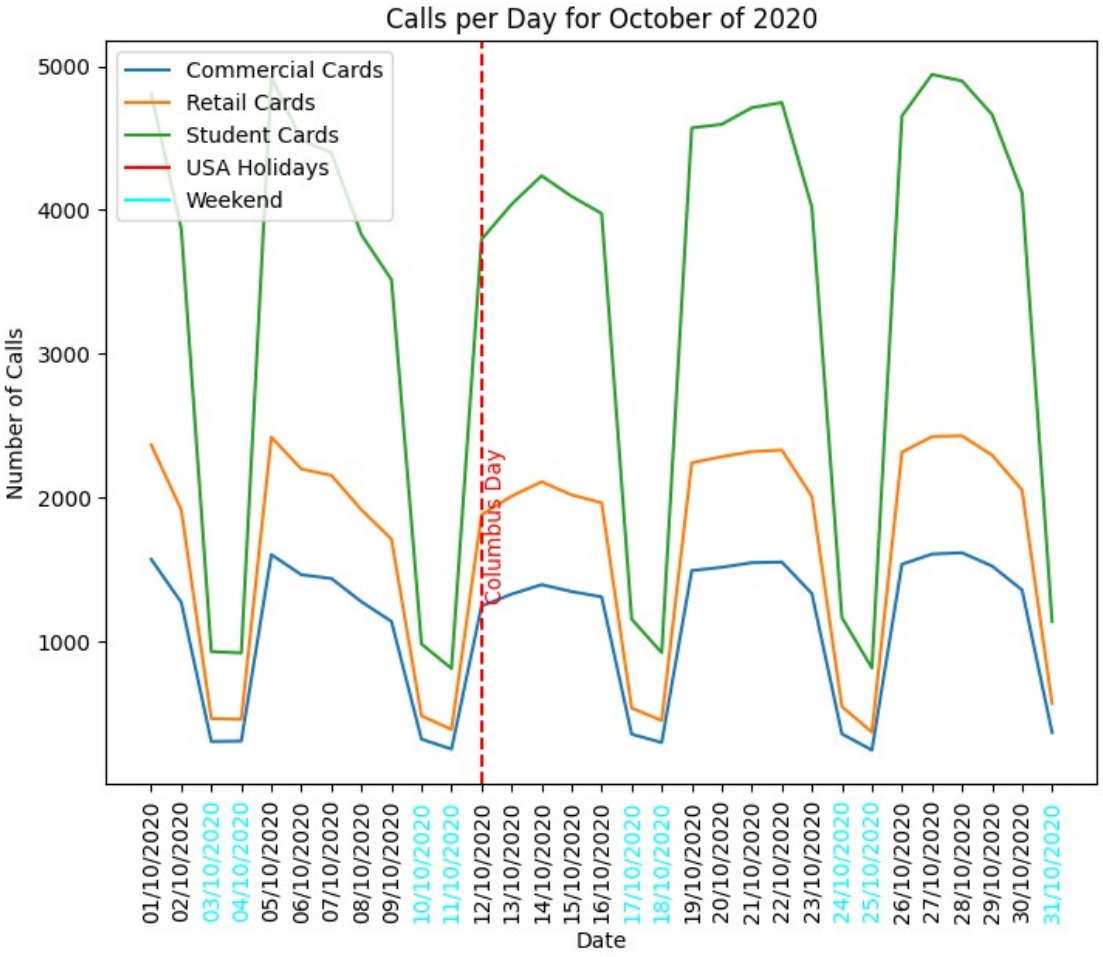

Calls per Day for May of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Memorial Day
Number of Calls
0
1000
2000
3000
4000
5000
Date
01/05/2020
02/05/2020
03/05/2020
04/05/2020
05/05/2020
06/05/2020
07/05/2020
08/05/2020
09/05/2020
10/05/2020
11/05/2020
12/05/2020
13/05/2020
14/05/2020
15/05/2020
16/05/2020
17/05/2020
18/05/2020
19/05/2020
20/05/2020
21/05/2020
22/05/2020
23/05/2020
24/05/2020
25/05/2020
26/05/2020
27/05/2020
28/05/2020
29/05/2020
30/05/2020
31/05/2020

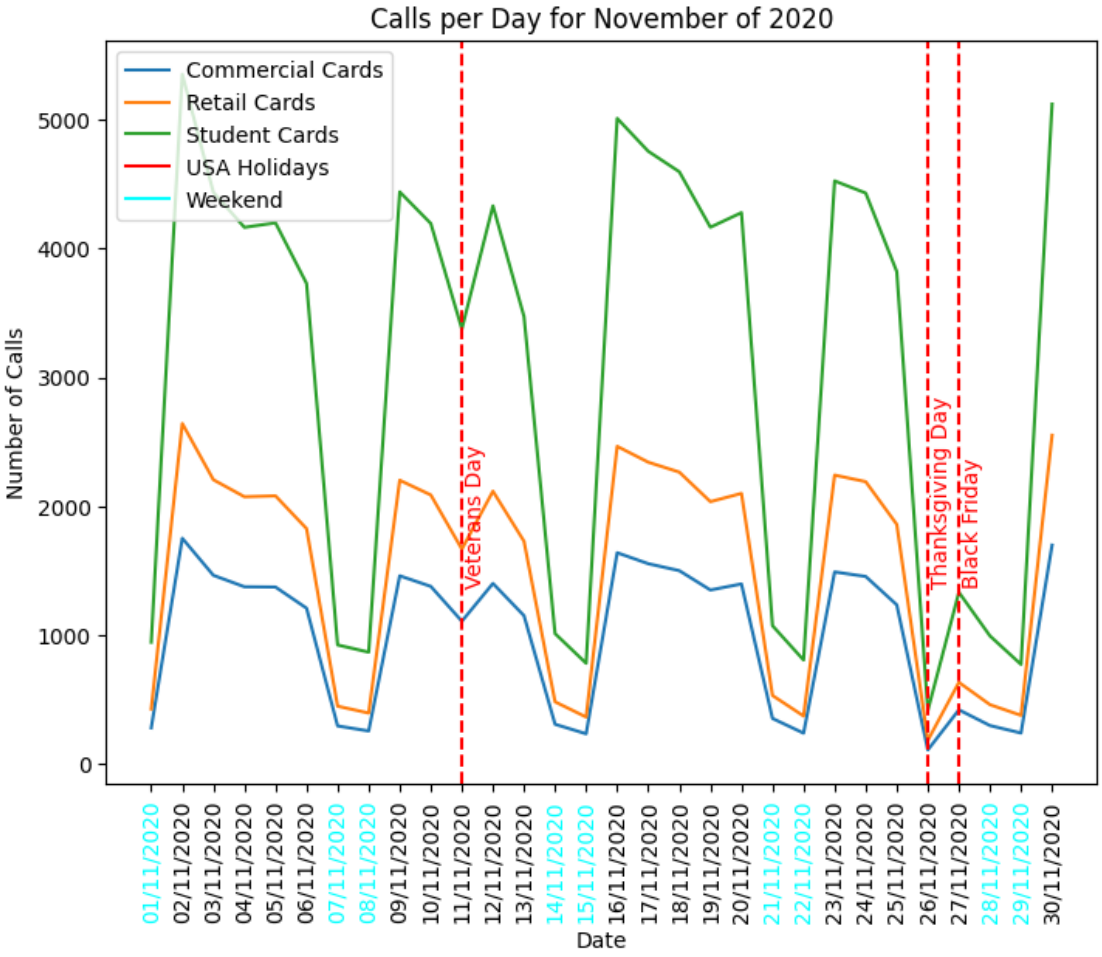

Calls per Day for June of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
01/06/2020
02/06/2020
03/06/2020
04/06/2020
05/06/2020
06/06/2020
07/06/2020
08/06/2020
09/06/2020
10/06/2020
11/06/2020
12/06/2020
13/06/2020
14/06/2020
15/06/2020
16/06/2020
17/06/2020
18/06/2020
19/06/2020
20/06/2020
21/06/2020
22/06/2020
23/06/2020
24/06/2020
25/06/2020
26/06/2020
27/06/2020
28/06/2020
29/06/2020
30/06/2020
Date

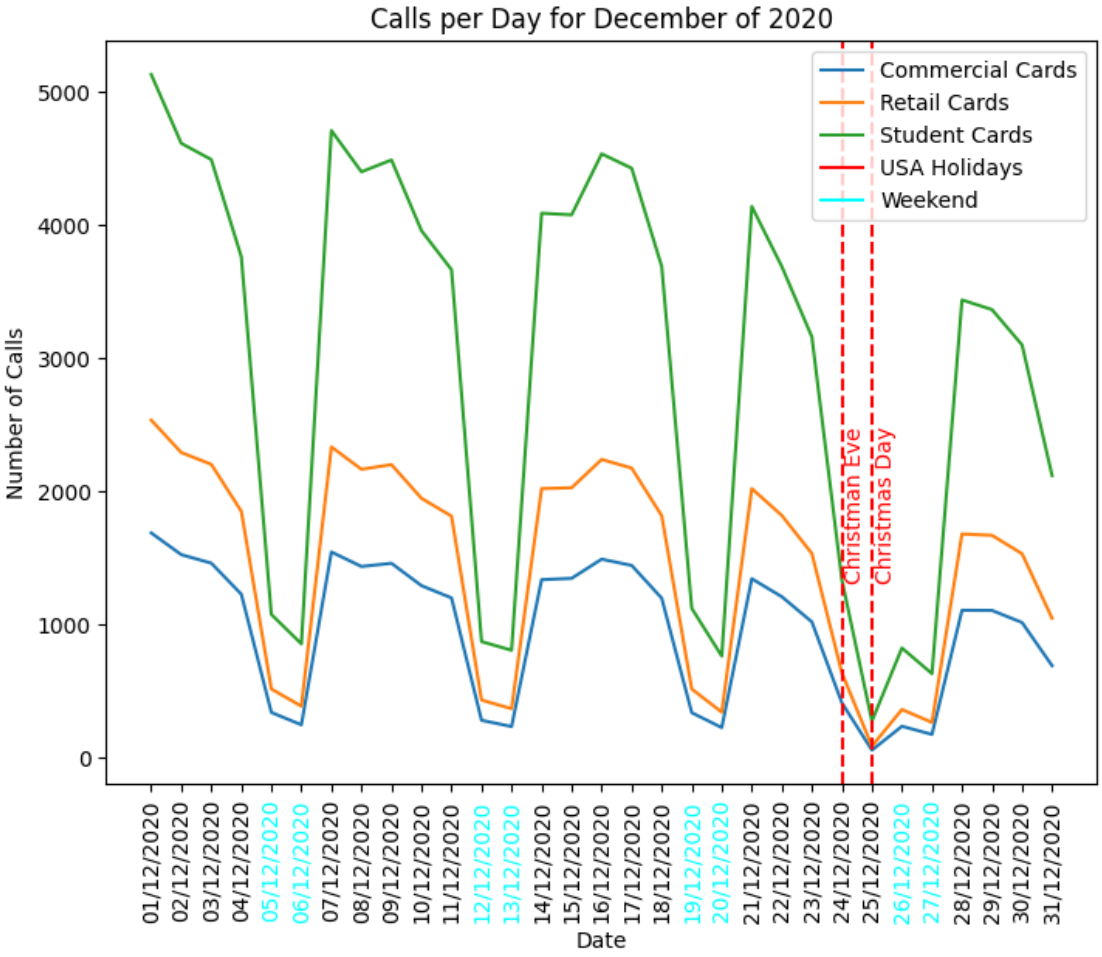

Calls per Day for July of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Independence Day
Number of Calls
0
1000
2000
3000
4000
01/07/2020
02/07/2020
03/07/2020
04/07/2020
05/07/2020
06/07/2020
07/07/2020
08/07/2020
09/07/2020
10/07/2020
11/07/2020
12/07/2020
13/07/2020
14/07/2020
15/07/2020
16/07/2020
17/07/2020
18/07/2020
19/07/2020
20/07/2020
21/07/2020
22/07/2020
23/07/2020
24/07/2020
25/07/2020
26/07/2020
27/07/2020
28/07/2020
29/07/2020
30/07/2020
31/07/2020
Date

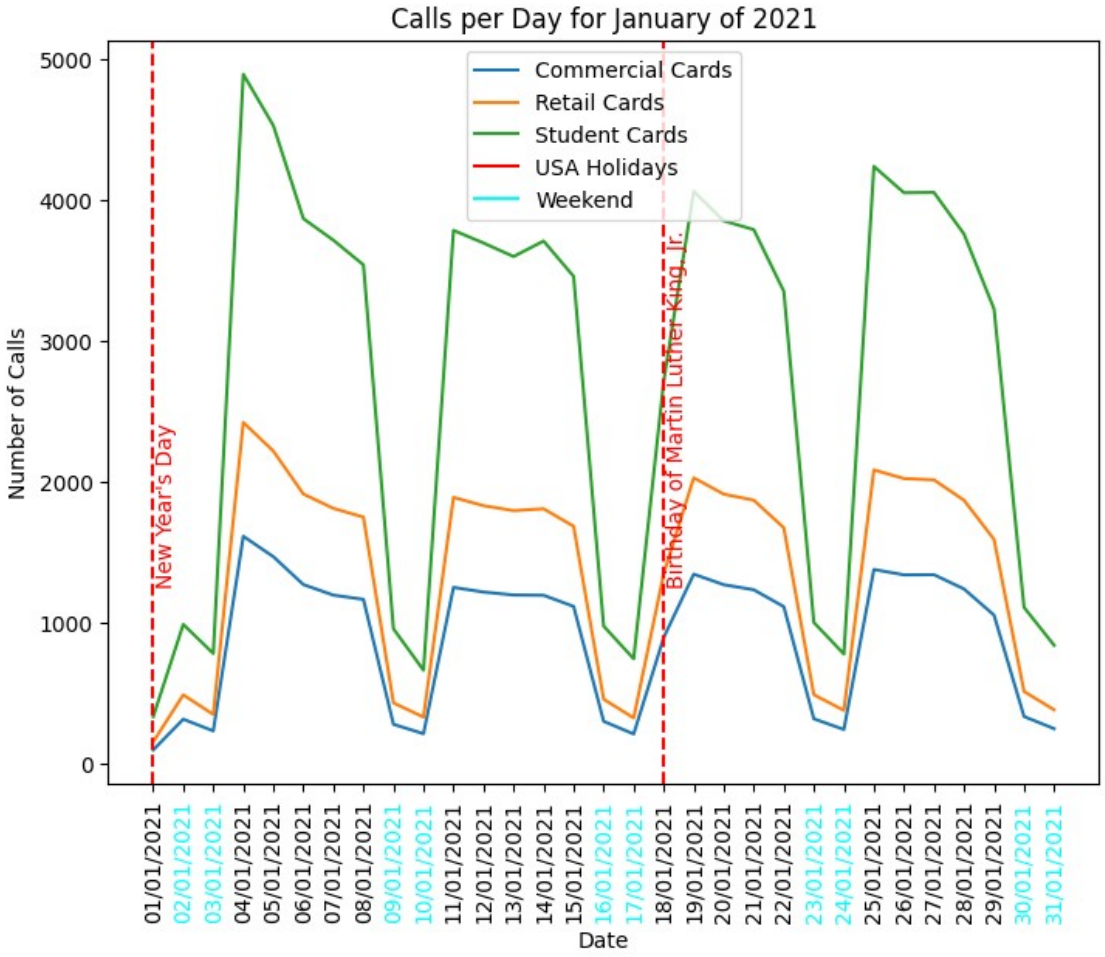

Calls per Day for August of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
Date
01/08/2020
02/08/2020
03/08/2020
04/08/2020
05/08/2020
06/08/2020
07/08/2020
08/08/2020
09/08/2020
10/08/2020
11/08/2020
12/08/2020
13/08/2020
14/08/2020
15/08/2020
16/08/2020
17/08/2020
18/08/2020
19/08/2020
20/08/2020
21/08/2020
22/08/2020
23/08/2020
24/08/2020
25/08/2020
26/08/2020
27/08/2020
28/08/2020
29/08/2020
30/08/2020
31/08/2020

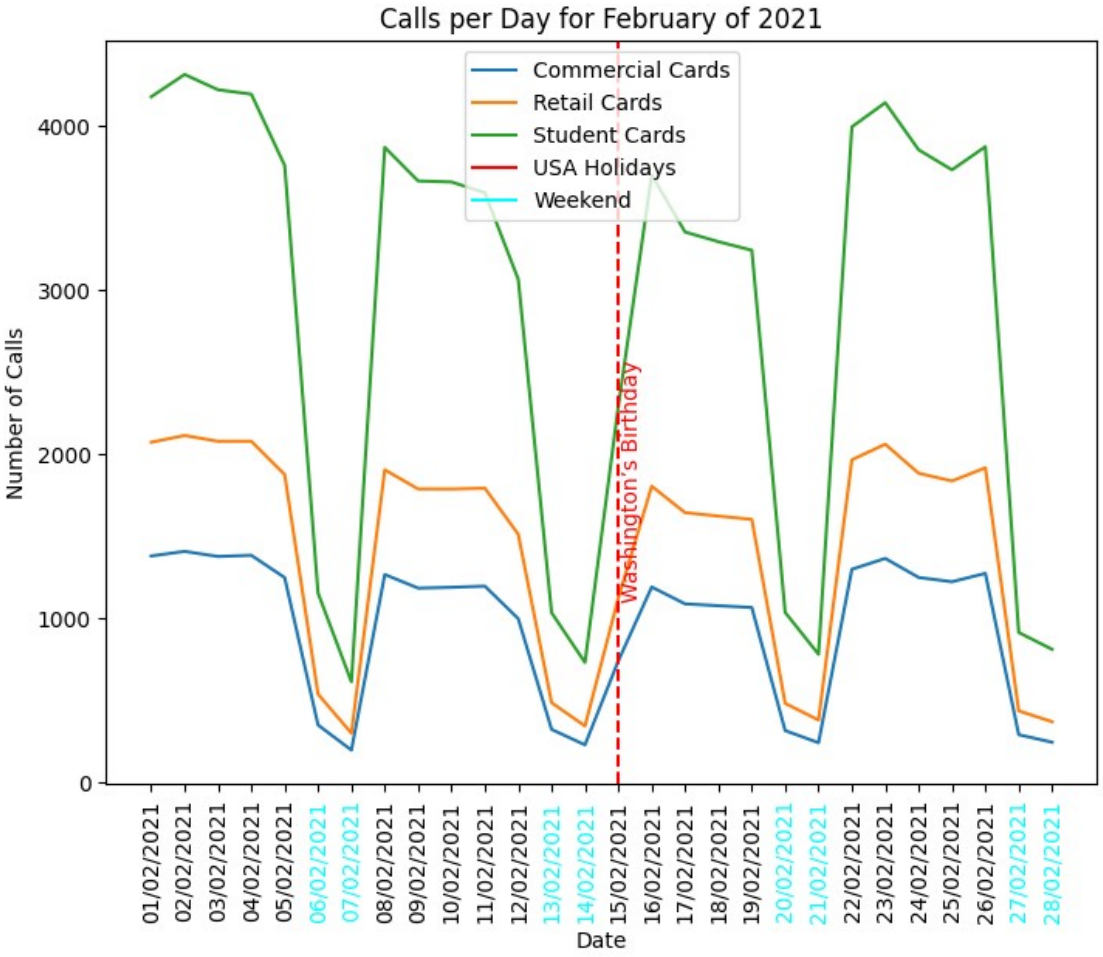

Calls per Day for September of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Labor Day
Number of Calls
Date
0
1000
2000
3000
4000
5000
01/09/2020
02/09/2020
03/09/2020
04/09/2020
05/09/2020
06/09/2020
07/09/2020
08/09/2020
09/09/2020
10/09/2020
11/09/2020
12/09/2020
13/09/2020
14/09/2020
15/09/2020
16/09/2020
17/09/2020
18/09/2020
19/09/2020
20/09/2020
21/09/2020
22/09/2020
23/09/2020
24/09/2020
25/09/2020
26/09/2020
27/09/2020
28/09/2020
29/09/2020
30/09/2020

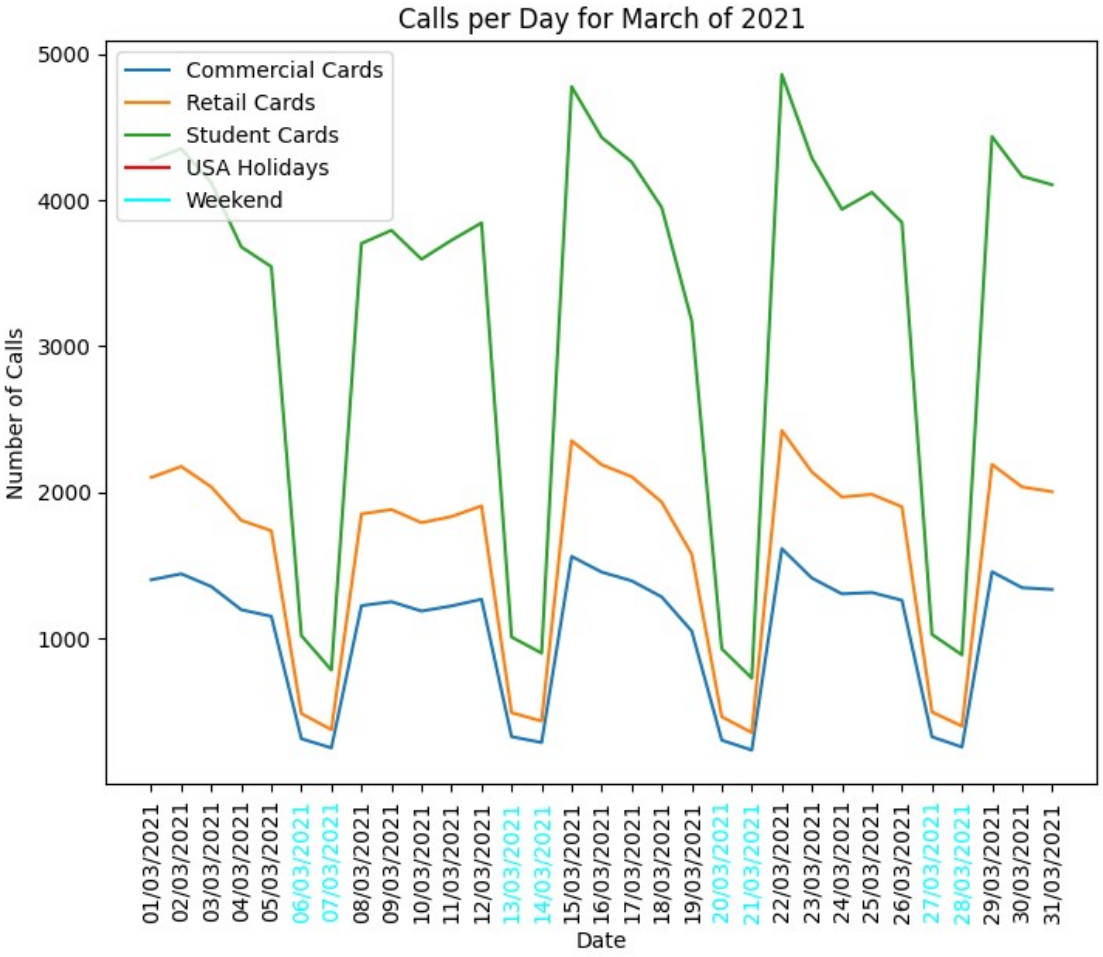

Calls per Day for October of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Columbus Day
Number of Calls
5000
4000
3000
2000
1000
01/10/2020
02/10/2020
03/10/2020
04/10/2020
05/10/2020
06/10/2020
07/10/2020
08/10/2020
09/10/2020
10/10/2020
11/10/2020
12/10/2020
13/10/2020
14/10/2020
15/10/2020
16/10/2020
17/10/2020
18/10/2020
19/10/2020
20/10/2020
21/10/2020
22/10/2020
23/10/2020
24/10/2020
25/10/2020
26/10/2020
27/10/2020
28/10/2020
29/10/2020
30/10/2020
31/10/2020
Date

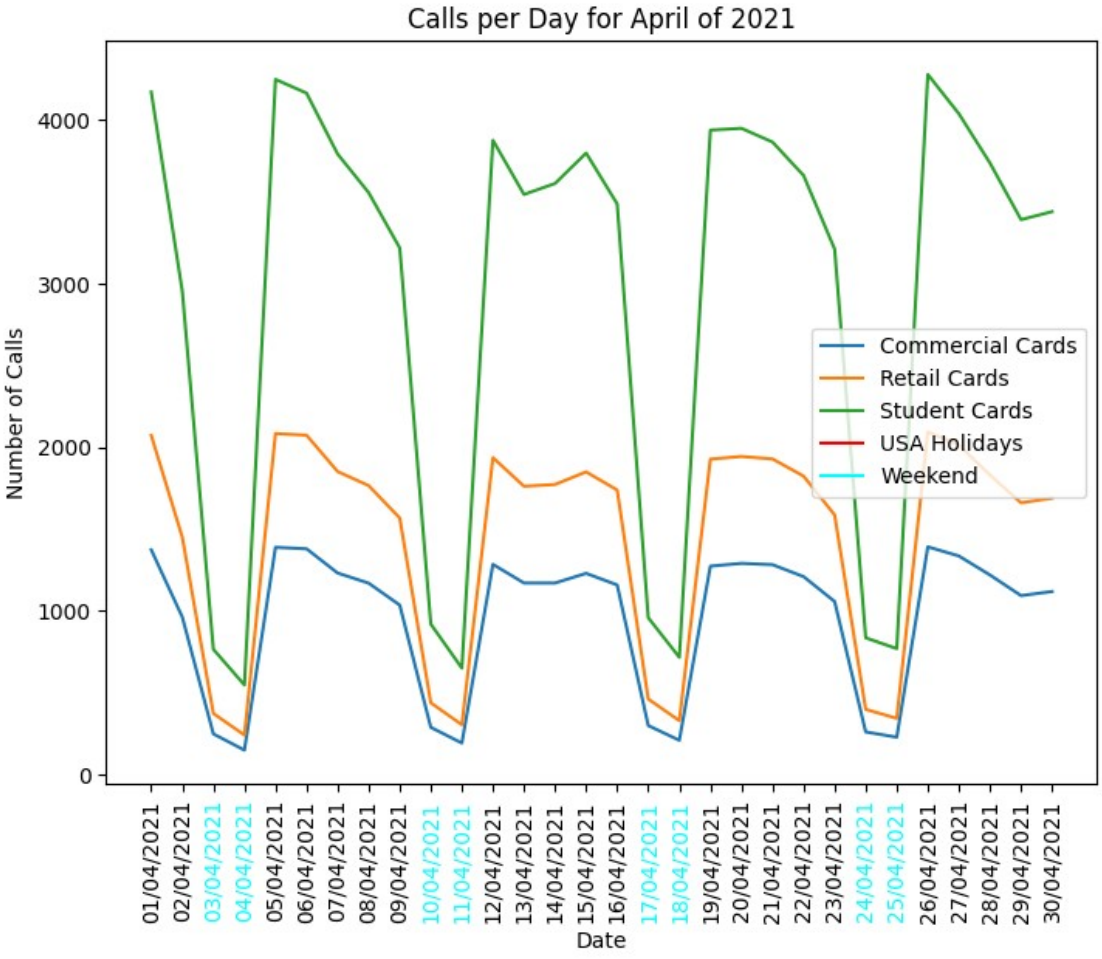

Calls per Day for November of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
Veterans Day
Thanksgiving Day
Black Friday
01/11/2020
02/11/2020
03/11/2020
04/11/2020
05/11/2020
06/11/2020
07/11/2020
08/11/2020
09/11/2020
10/11/2020
11/11/2020
12/11/2020
13/11/2020
14/11/2020
15/11/2020
16/11/2020
17/11/2020
18/11/2020
19/11/2020
20/11/2020
21/11/2020
22/11/2020
23/11/2020
24/11/2020
25/11/2020
26/11/2020
27/11/2020
28/11/2020
29/11/2020
30/11/2020
Date


Calls per Day for December of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
5000
4000
3000
2000
1000
0
Christman Eve
Christmas Day
01/12/2020
02/12/2020
03/12/2020
04/12/2020
05/12/2020
06/12/2020
07/12/2020
08/12/2020
09/12/2020
10/12/2020
11/12/2020
12/12/2020
13/12/2020
14/12/2020
15/12/2020
16/12/2020
17/12/2020
18/12/2020
19/12/2020
20/12/2020
21/12/2020
22/12/2020
23/12/2020
24/12/2020
25/12/2020
26/12/2020
27/12/2020
28/12/2020
29/12/2020
30/12/2020
31/12/2020
Date


Calls per Day for January of 2021
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
New Year's Day
Birthday of Martin Luther King, Jr.
Number of Calls
0
1000
2000
3000
4000
5000
01/01/2021
02/01/2021
03/01/2021
04/01/2021
05/01/2021
06/01/2021
07/01/2021
08/01/2021
09/01/2021
10/01/2021
11/01/2021
12/01/2021
13/01/2021
14/01/2021
15/01/2021
16/01/2021
17/01/2021
18/01/2021
19/01/2021
20/01/2021
21/01/2021
22/01/2021
23/01/2021
24/01/2021
25/01/2021
26/01/2021
27/01/2021
28/01/2021
29/01/2021
30/01/2021
31/01/2021
Date


Calls per Day for February of 2021
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Washington's Birthday
Number of Calls
0
1000
2000
3000
4000
01/02/2021
02/02/2021
03/02/2021
04/02/2021
05/02/2021
06/02/2021
07/02/2021
08/02/2021
09/02/2021
10/02/2021
11/02/2021
12/02/2021
13/02/2021
14/02/2021
15/02/2021
16/02/2021
17/02/2021
18/02/2021
19/02/2021
20/02/2021
21/02/2021
22/02/2021
23/02/2021
24/02/2021
25/02/2021
26/02/2021
27/02/2021
28/02/2021
Date

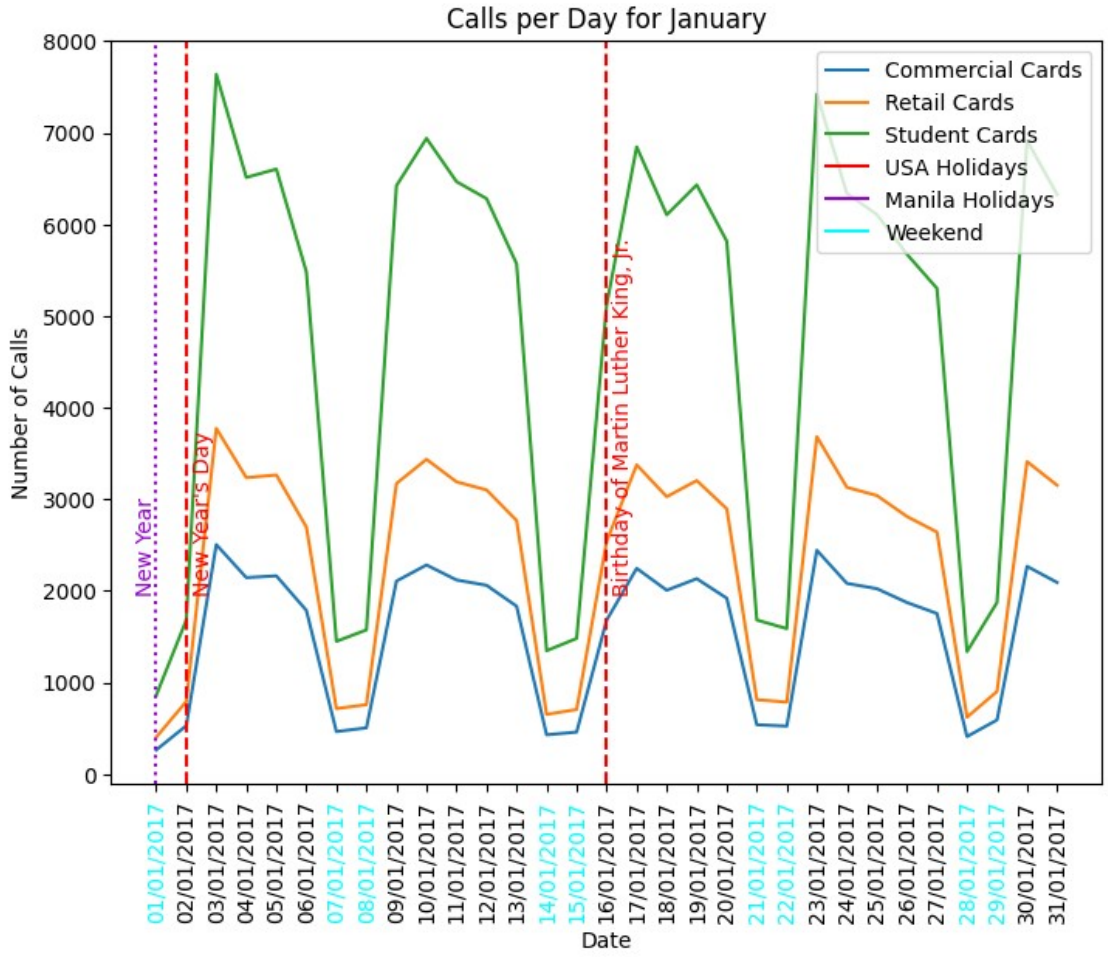

Calls per Day for March of 2021
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Weekend
Number of Calls
5000
4000
3000
2000
1000
01/03/2021
02/03/2021
03/03/2021
04/03/2021
05/03/2021
06/03/2021
07/03/2021
08/03/2021
09/03/2021
10/03/2021
11/03/2021
12/03/2021
13/03/2021
14/03/2021
15/03/2021
16/03/2021
17/03/2021
18/03/2021
19/03/2021
20/03/2021
21/03/2021
22/03/2021
23/03/2021
24/03/2021
25/03/2021
26/03/2021
27/03/2021
28/03/2021
29/03/2021
30/03/2021
31/03/2021
Date

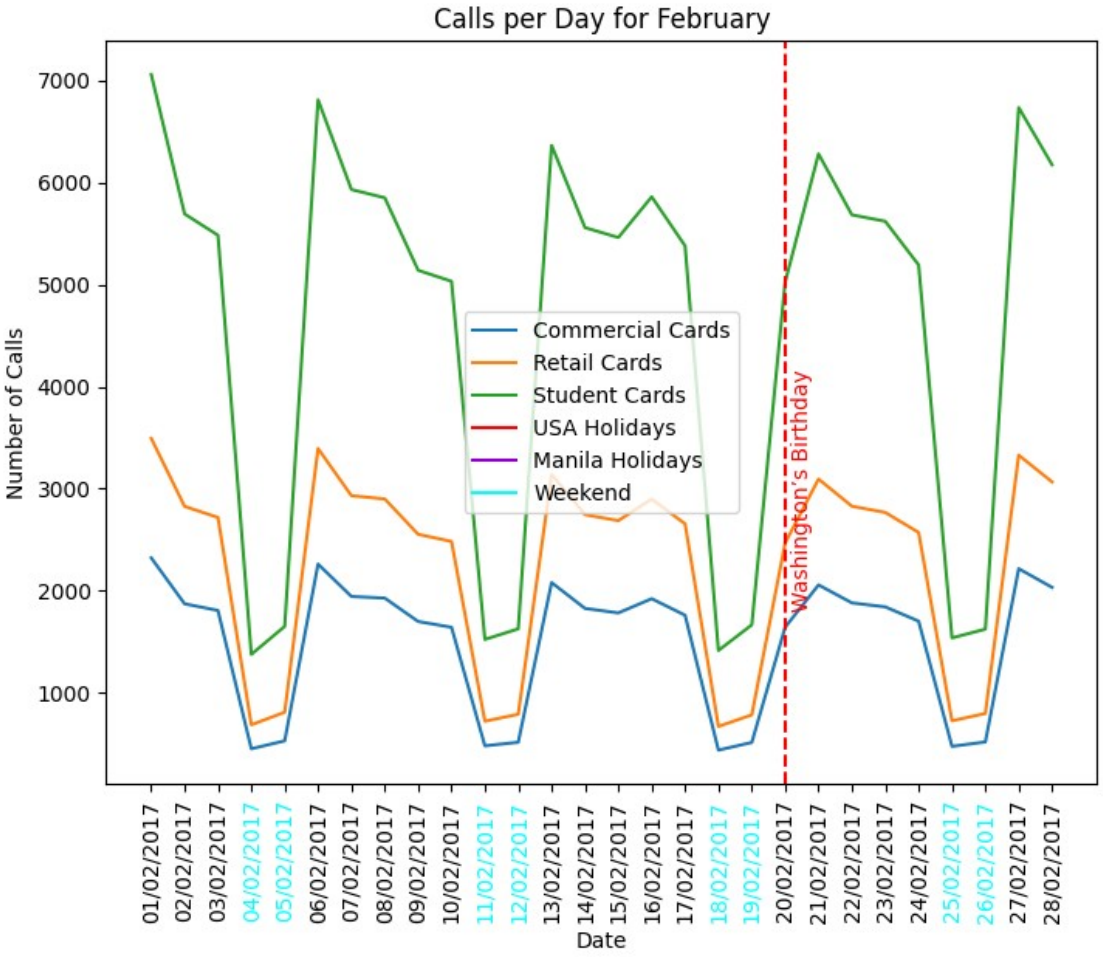


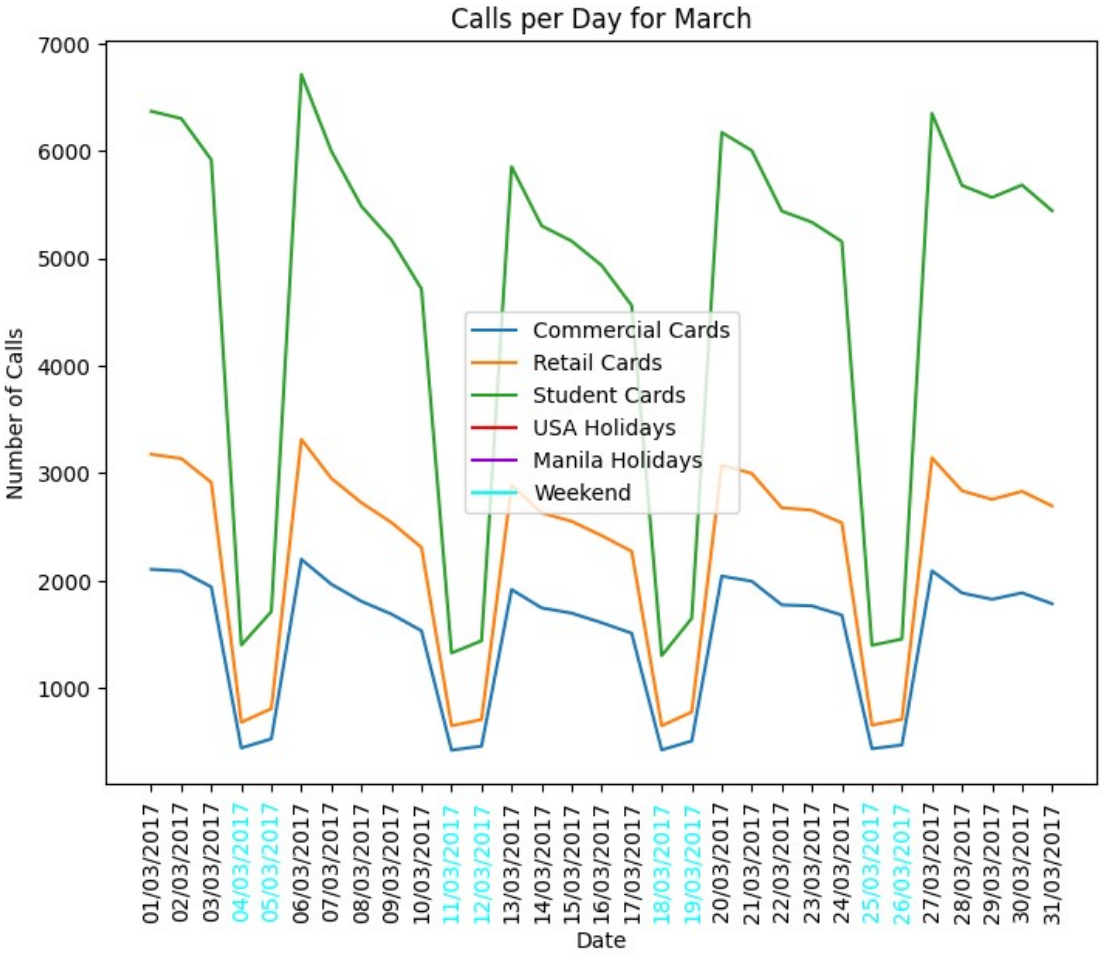

```
# Manila Holiday Calendar
class idcalen(AbstractHolidayCalendar):
    rules=[
        Holiday("New Year",month=1,day=1),
        Holiday("Day of Valor",month=4,day=9),
        Holiday("Labor Day",month=5,day=1),
        Holiday("Independence Day",month=6,day=12),
        Holiday("National heroes
day",month=8,day=31,offset=pd.DateOffset(weekday=MO(-1))),
        Holiday("Bonifacio",month=11,day=30),
        Holiday("Christmas",month=12,day=25),
        Holiday("Rizal",month=12,day=30),
        Holiday("Maundy Thursday",month=4,day=13,year=2017),
        Holiday("Maundy Thursday",month=3,day=29,year=2018),
        Holiday("Maundy Thursday",month=4,day=18,year=2019),
        Holiday("Maundy Thursday",month=4,day=9,year=2020),
        Holiday("Maundy Thursday",month=4,day=1,year=2021),
        Holiday("Good Friday",month=4,day=14,year=2017),
        Holiday("Good Friday",month=3,day=30,year=2018),
        Holiday("Good Friday",month=4,day=19,year=2019),
        Holiday("Good Friday",month=4,day=10,year=2020),
```

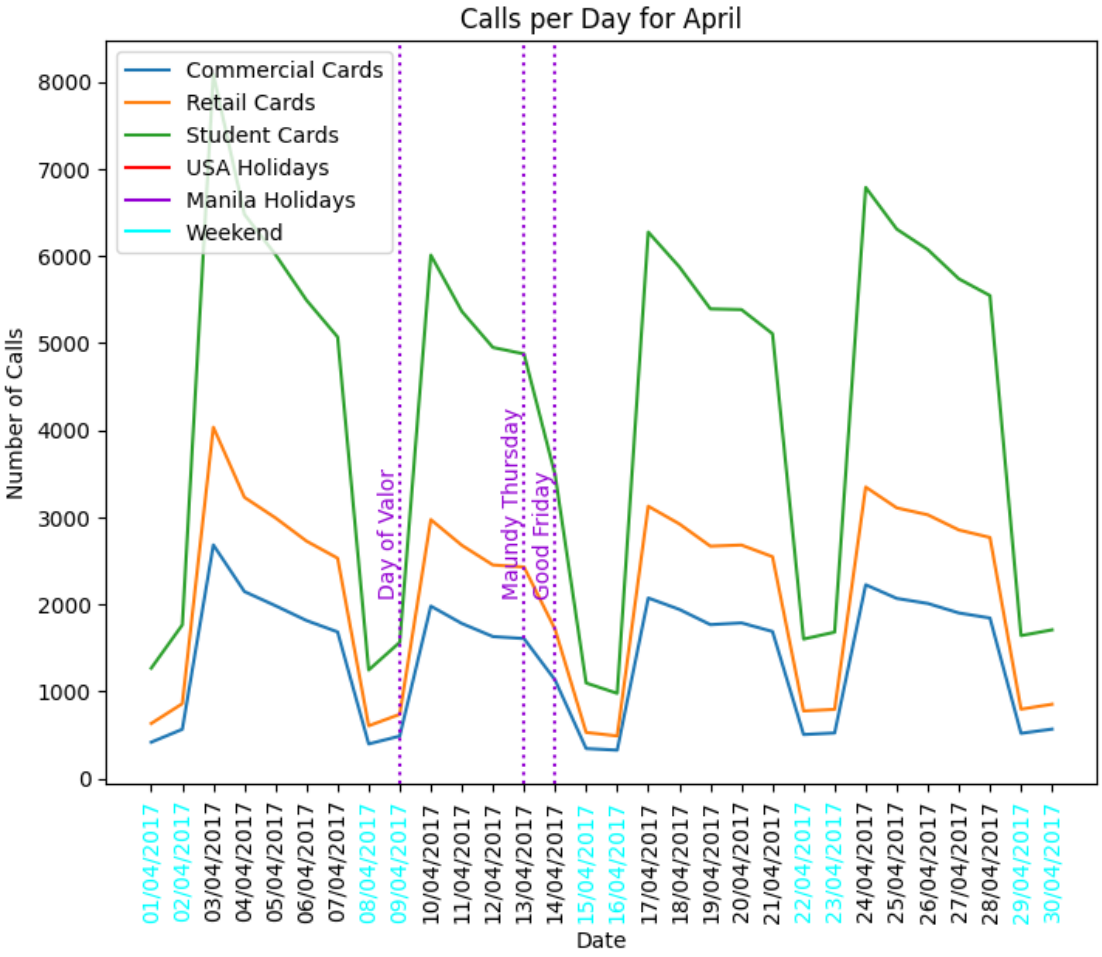
```
        Holiday("Good Friday",month=4,day=2,year=2021)
    ]

manila_cal=idcalen()
manila_holidays = manila_cal.holidays(start="2017-01-01",end="2021-04-
30",return_name=True)
m_holidays = manila_holidays.to_frame("name")
m_holidays.reset_index(inplace=True)
m_holidays.rename(columns={"index":"date"}, inplace=True)

# 2017 data for each month
# Low number of calls on weekends
# Low number of calls on Big Holidays
holidays_2017 = holidays[[True if dt.year == 2017 else False for dt in
holidays.date]]
m_holidays_2017 = m_holidays[[True if dt.year == 2017 else False for
dt in m_holidays.date]]
months = df_17.month.unique()
for m in months:
    x = range((df_17.month == m).sum())
    cc_values = df_17[df_17.month == m].cc
    rc_values = df_17[df_17.month == m].rc
    sc_values = df_17[df_17.month == m].sc
    month_holidays = holidays_2017[[True if dt.month == m else False
for dt in holidays_2017.date]]
    monhol_values = [(dt.date.day-1, dt['name']) for i, dt in
month_holidays.iterrows()]
    m_month_holidays = m_holidays_2017[[True if dt.month == m else
False for dt in m_holidays_2017.date]]
    m_monhol_values = [(dt.date.day-1, dt['name']) for i, dt in
m_month_holidays.iterrows()]
    fig = plt.figure()
    ax = fig.add_axes([1,1,1,1])
    ax.plot(x, cc_values, label='Commercial Cards')
    ax.plot(x, rc_values, label='Retail Cards')
    ax.plot(x, sc_values, label='Student Cards')
    ax.plot([], [], label='USA Holidays', c='red')
    ax.plot([], [], label='Manila Holidays', c='darkviolet')
    ax.plot([], [], label='Weekend', c='cyan')
    for mhd, mhn in monhol_values:
        if mhn == 'Thanksgiving Day':
            ax.axvline(mhd+1, c='red', ls='--')
            ax.text(mhd+1.1, ax.get_ylim()[1]//4, "Black Friday",
rotation=90, c='red')
        if mhn == 'Christmas Day':
            ax.axvline(mhd-1, c='red', ls='--')
            ax.text(mhd-0.9, ax.get_ylim()[1]//4, "Christman Eve",
rotation=90, c='red')
        ax.axvline(mhd, c='red', ls='--')
        ax.text(mhd+0.2, ax.get_ylim()[1]//4, mhn, rotation=90,
c='red')
```

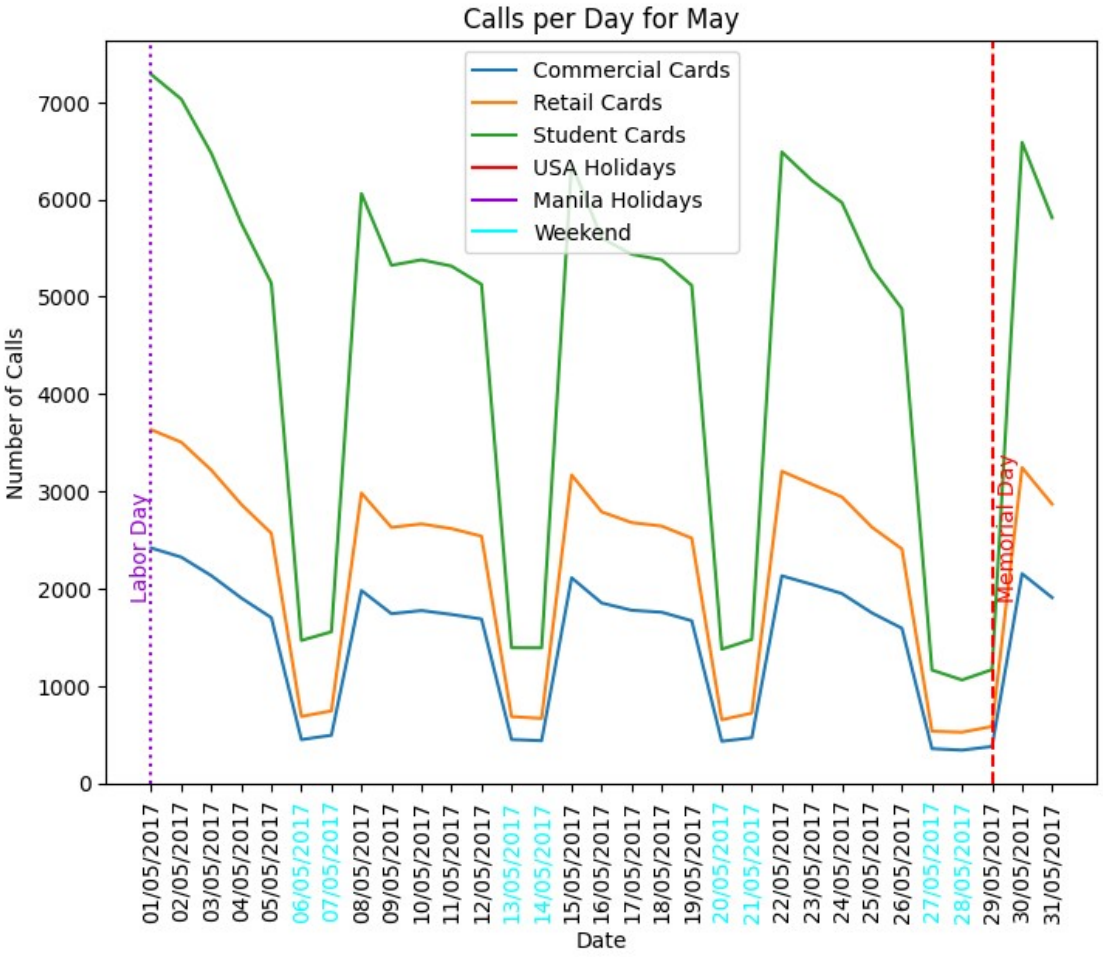
```
    for mhd, mhn in m_monhol_values:
        ax.axvline(mhd, c='darkviolet', ls=':')
        ax.text(mhd-0.7, ax.get_ylim()[1]//4, mhn, rotation=90,
c='darkviolet')
    ax.set_xticks(x, df_17[df_17.month == m].date, rotation=90)
    for i, dt in enumerate(df_17[df_17.month == m].date):
        if arrow.get(dt, "DD/MM/YYYY").format("ddd") in ["Sat",
"Sun"]:
            ax.get_xticklabels()[i].set_color('cyan')
    ax.set_xlabel("Date")
    ax.set_ylabel("Number of Calls")
    ax.set_title(f"Calls per Day for {arrow.get(str(m),
'M').format('MMMM')}")
    ax.legend()
    plt.show()
```

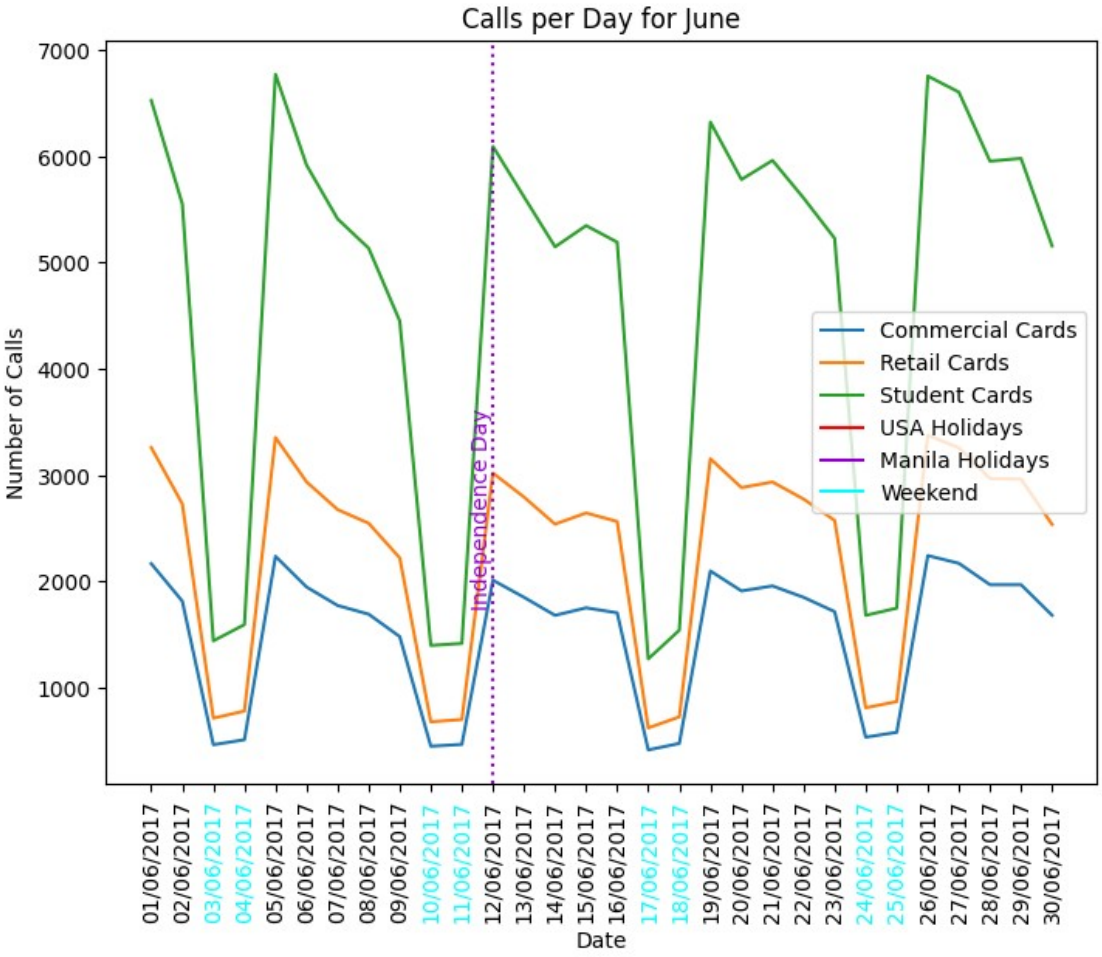


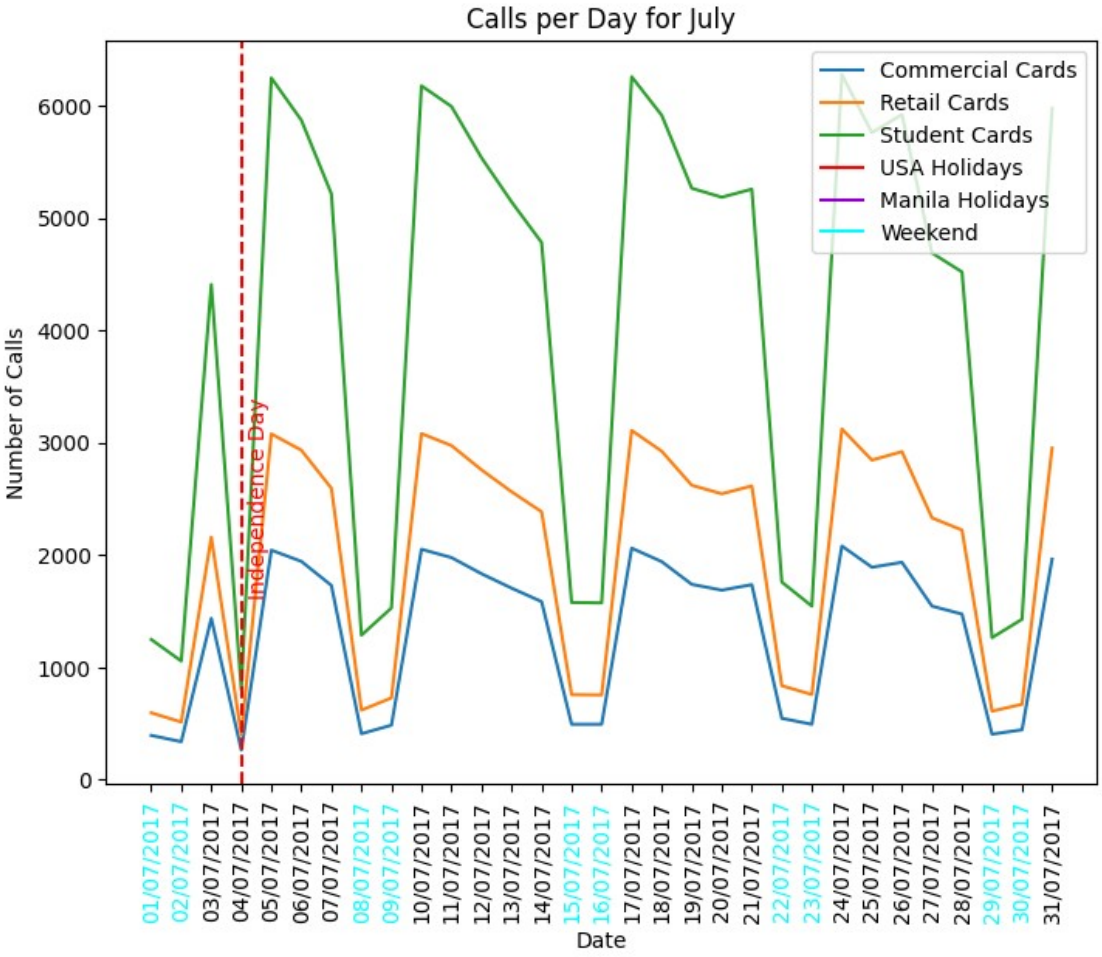

Calls per Day for February
Number of Calls
7000
6000
5000
4000
3000
2000
1000
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Washington's Birthday
01/02/2017
02/02/2017
03/02/2017
04/02/2017
05/02/2017
06/02/2017
07/02/2017
08/02/2017
09/02/2017
10/02/2017
11/02/2017
12/02/2017
13/02/2017
14/02/2017
15/02/2017
16/02/2017
17/02/2017
18/02/2017
19/02/2017
20/02/2017
21/02/2017
22/02/2017
23/02/2017
24/02/2017
25/02/2017
26/02/2017
27/02/2017
28/02/2017
Date

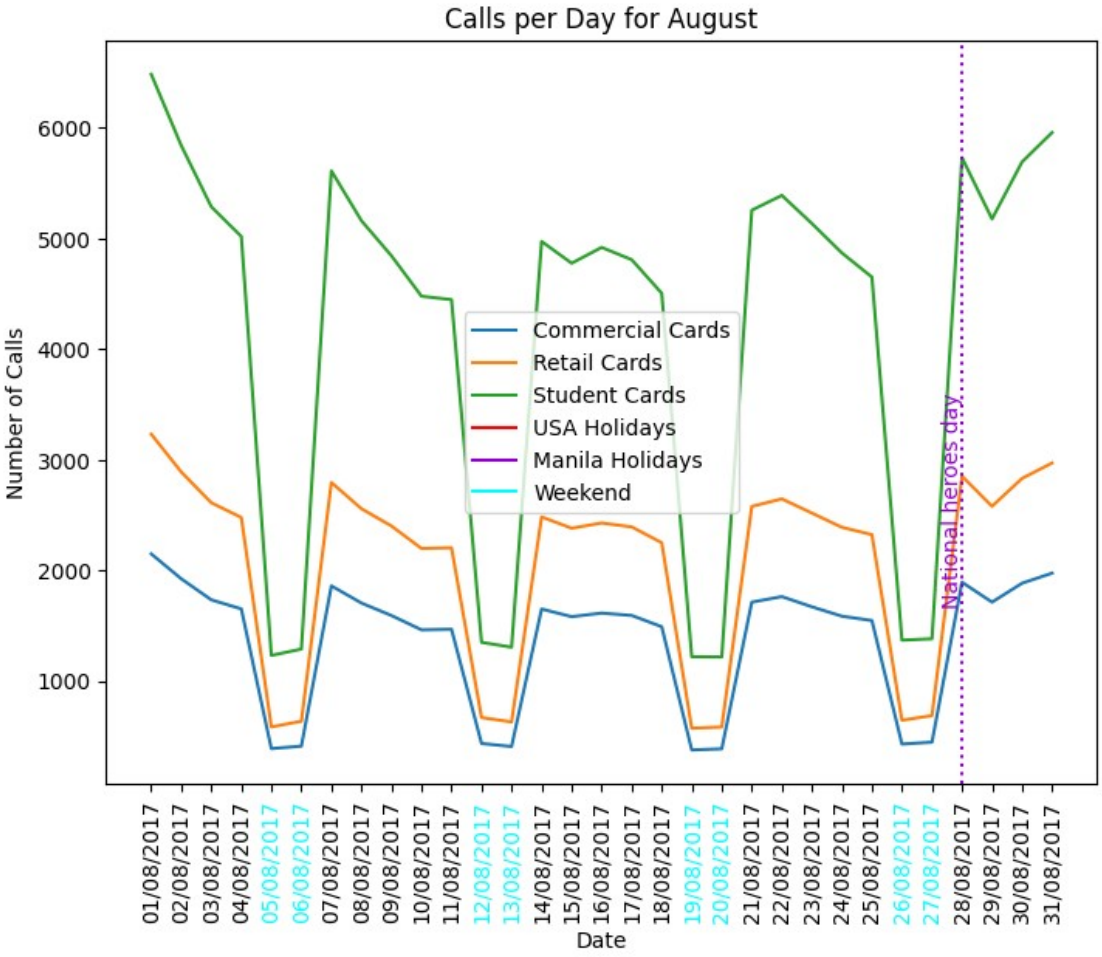

Calls per Day for March
Number of Calls
7000
6000
5000
4000
3000
2000
1000
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
01/03/2017
02/03/2017
03/03/2017
04/03/2017
05/03/2017
06/03/2017
07/03/2017
08/03/2017
09/03/2017
10/03/2017
11/03/2017
12/03/2017
13/03/2017
14/03/2017
15/03/2017
16/03/2017
17/03/2017
18/03/2017
19/03/2017
20/03/2017
21/03/2017
22/03/2017
23/03/2017
24/03/2017
25/03/2017
26/03/2017
27/03/2017
28/03/2017
29/03/2017
30/03/2017
31/03/2017
Date

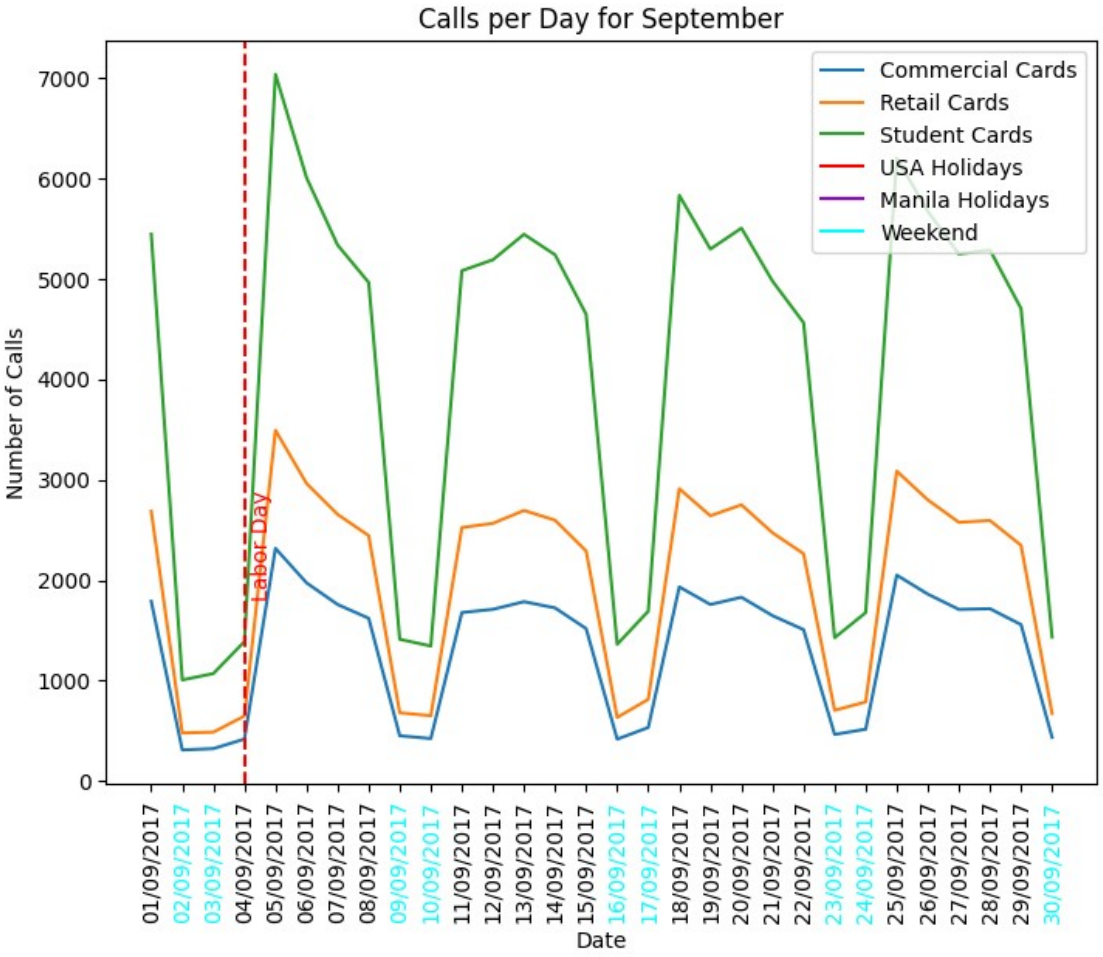

Calls per Day for April
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
8000
Day of Valor
Maundy Thursday
Good Friday
01/04/2017
02/04/2017
03/04/2017
04/04/2017
05/04/2017
06/04/2017
07/04/2017
08/04/2017
09/04/2017
10/04/2017
11/04/2017
12/04/2017
13/04/2017
14/04/2017
15/04/2017
16/04/2017
17/04/2017
18/04/2017
19/04/2017
20/04/2017
21/04/2017
22/04/2017
23/04/2017
24/04/2017
25/04/2017
26/04/2017
27/04/2017
28/04/2017
29/04/2017
30/04/2017
Date

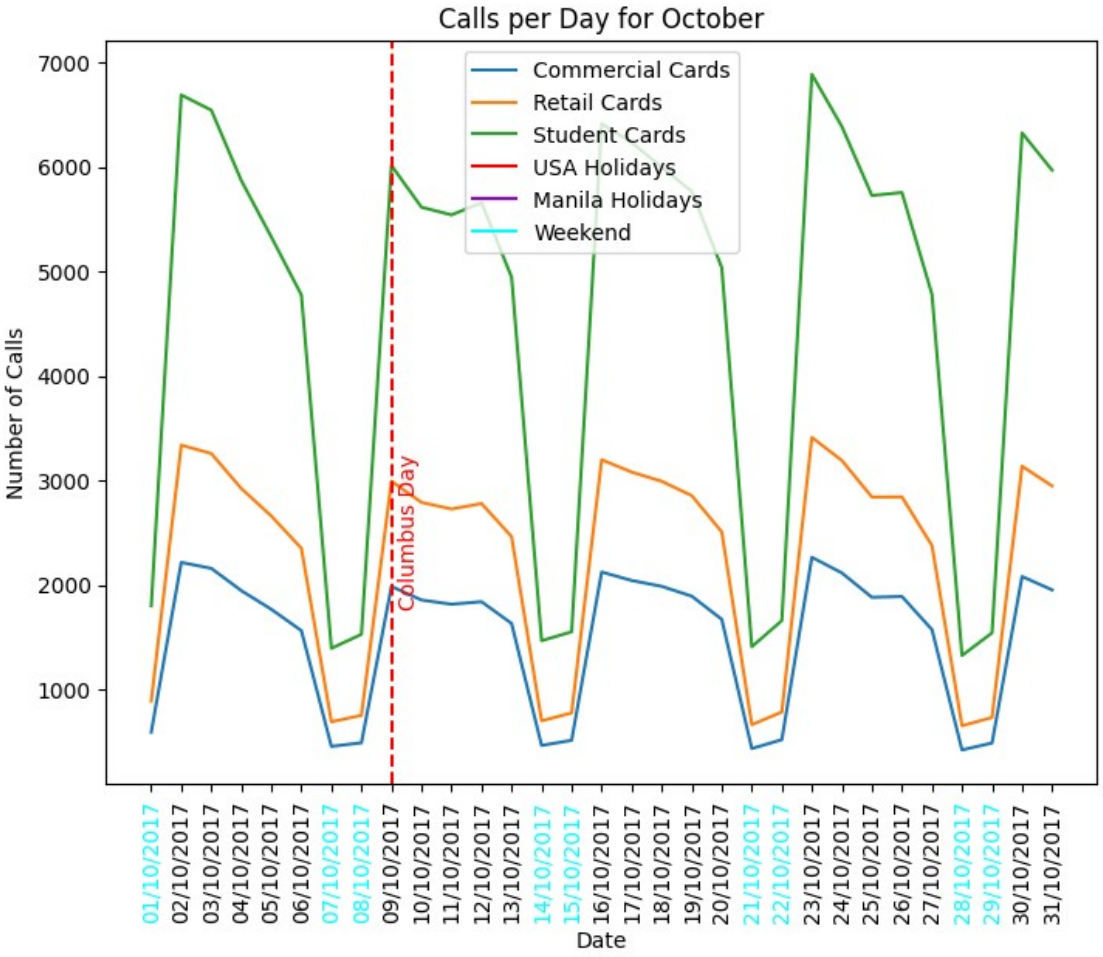

Calls per Day for May
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
Labor Day
Memorial Day
01/05/2017
02/05/2017
03/05/2017
04/05/2017
05/05/2017
06/05/2017
07/05/2017
08/05/2017
09/05/2017
10/05/2017
11/05/2017
12/05/2017
13/05/2017
14/05/2017
15/05/2017
16/05/2017
17/05/2017
18/05/2017
19/05/2017
20/05/2017
21/05/2017
22/05/2017
23/05/2017
24/05/2017
25/05/2017
26/05/2017
27/05/2017
28/05/2017
29/05/2017
30/05/2017
31/05/2017
Date

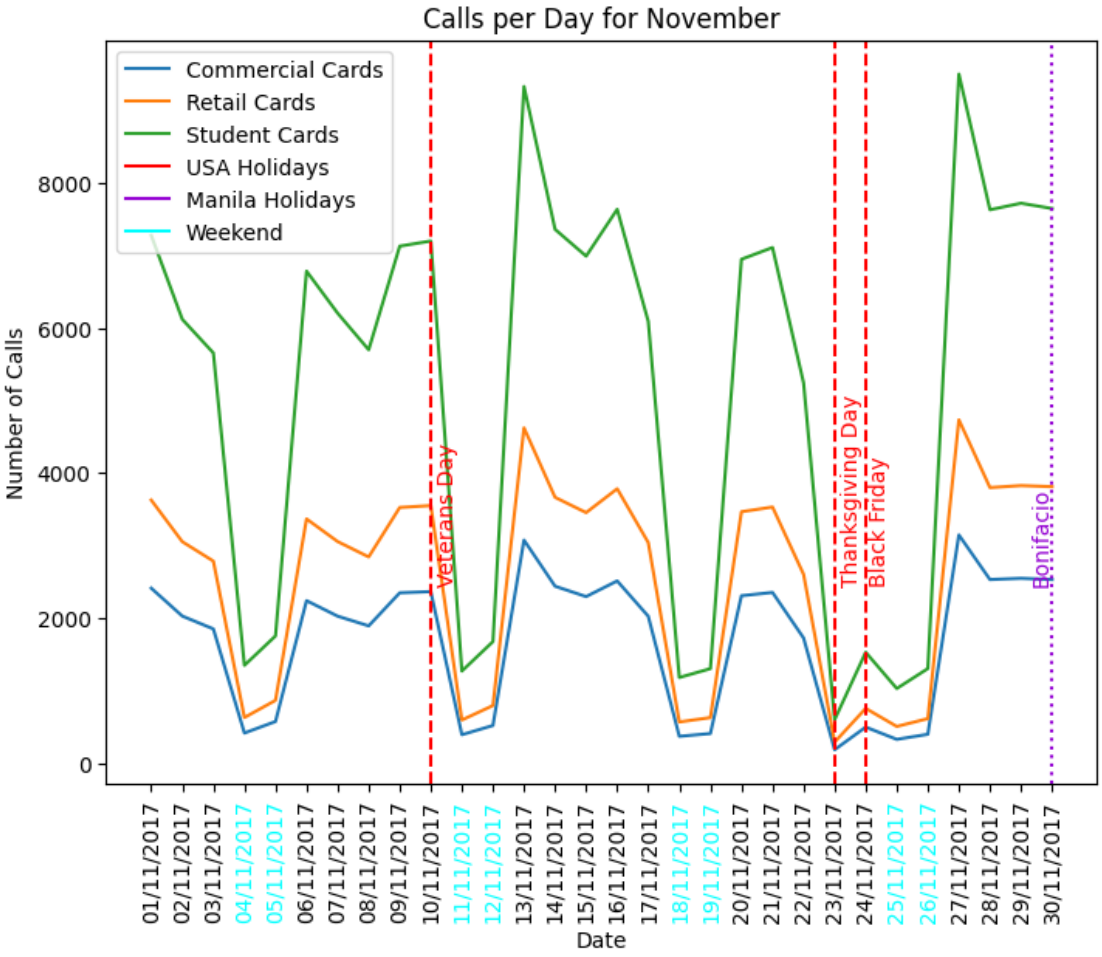

Calls per Day for June
Number of Calls
7000
6000
5000
4000
3000
2000
1000
Independence Day
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
01/06/2017
02/06/2017
03/06/2017
04/06/2017
05/06/2017
06/06/2017
07/06/2017
08/06/2017
09/06/2017
10/06/2017
11/06/2017
12/06/2017
13/06/2017
14/06/2017
15/06/2017
16/06/2017
17/06/2017
18/06/2017
19/06/2017
20/06/2017
21/06/2017
22/06/2017
23/06/2017
24/06/2017
25/06/2017
26/06/2017
27/06/2017
28/06/2017
29/06/2017
30/06/2017
Date

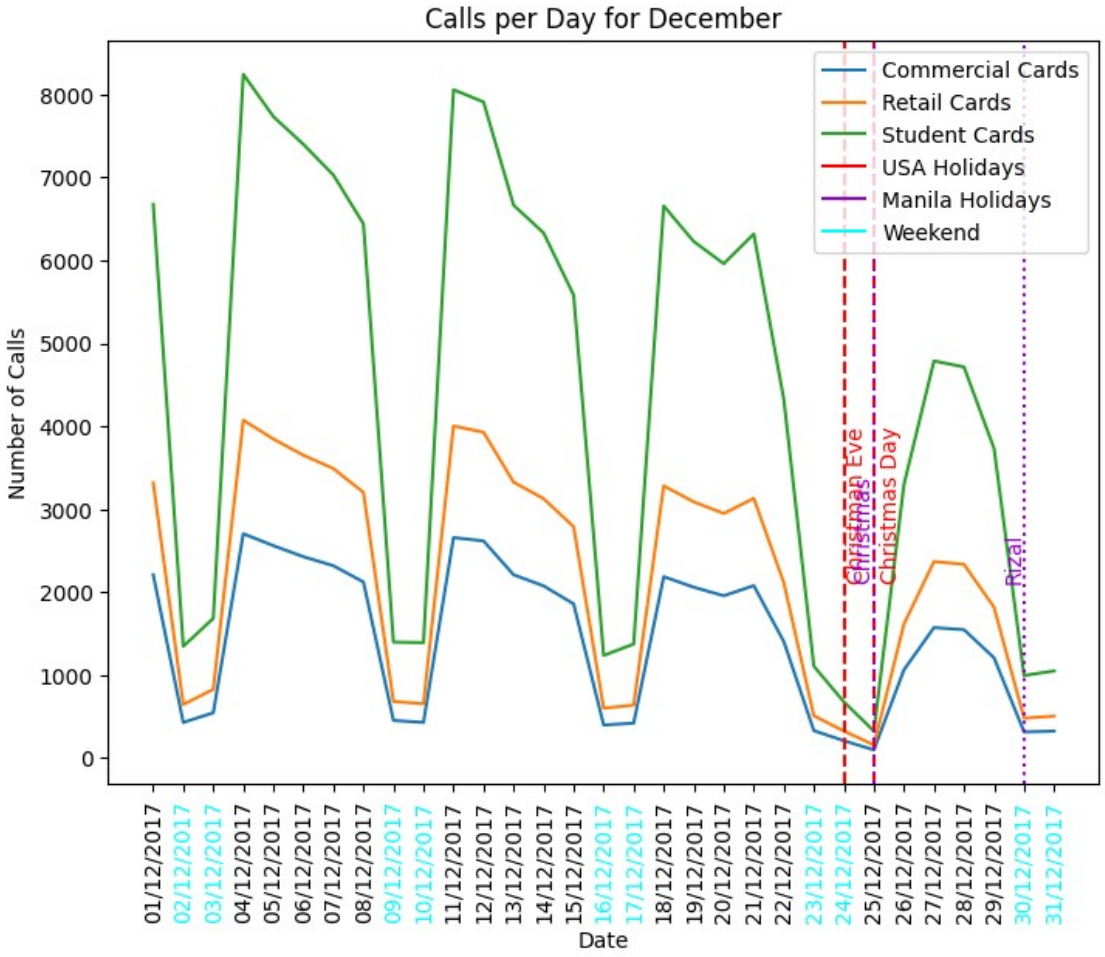

Calls per Day for July
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
Independence Day
01/07/2017
02/07/2017
03/07/2017
04/07/2017
05/07/2017
06/07/2017
07/07/2017
08/07/2017
09/07/2017
10/07/2017
11/07/2017
12/07/2017
13/07/2017
14/07/2017
15/07/2017
16/07/2017
17/07/2017
18/07/2017
19/07/2017
20/07/2017
21/07/2017
22/07/2017
23/07/2017
24/07/2017
25/07/2017
26/07/2017
27/07/2017
28/07/2017
29/07/2017
30/07/2017
31/07/2017
Date


Calls per Day for August
Number of Calls
6000
5000
4000
3000
2000
1000
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
National heroes day
01/08/2017
02/08/2017
03/08/2017
04/08/2017
05/08/2017
06/08/2017
07/08/2017
08/08/2017
09/08/2017
10/08/2017
11/08/2017
12/08/2017
13/08/2017
14/08/2017
15/08/2017
16/08/2017
17/08/2017
18/08/2017
19/08/2017
20/08/2017
21/08/2017
22/08/2017
23/08/2017
24/08/2017
25/08/2017
26/08/2017
27/08/2017
28/08/2017
29/08/2017
30/08/2017
31/08/2017
Date


Calls per Day for September
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
Labor Day
Date

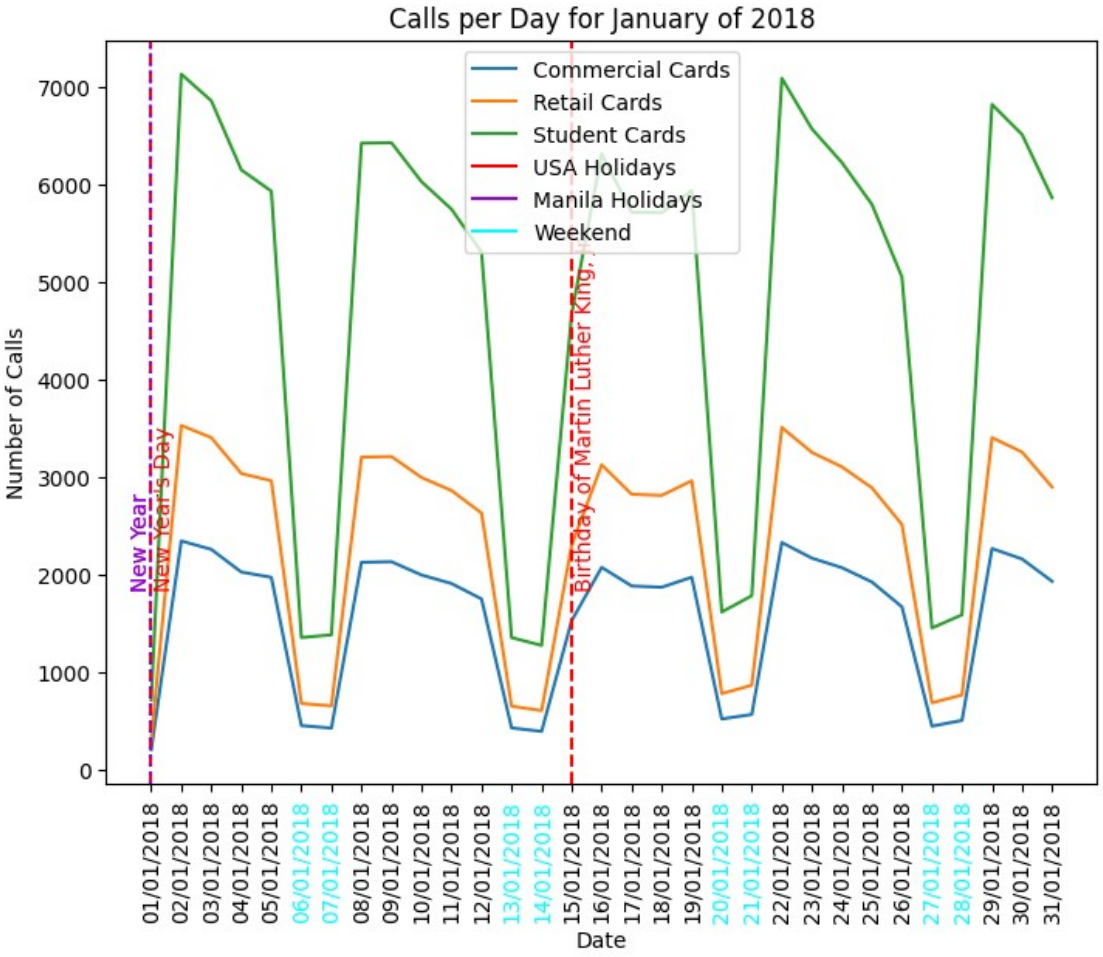

Calls per Day for October
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Columbus Day
Number of Calls
7000
6000
5000
4000
3000
2000
1000
01/10/2017
02/10/2017
03/10/2017
04/10/2017
05/10/2017
06/10/2017
07/10/2017
08/10/2017
09/10/2017
10/10/2017
11/10/2017
12/10/2017
13/10/2017
14/10/2017
15/10/2017
16/10/2017
17/10/2017
18/10/2017
19/10/2017
20/10/2017
21/10/2017
22/10/2017
23/10/2017
24/10/2017
25/10/2017
26/10/2017
27/10/2017
28/10/2017
29/10/2017
30/10/2017
31/10/2017
Date

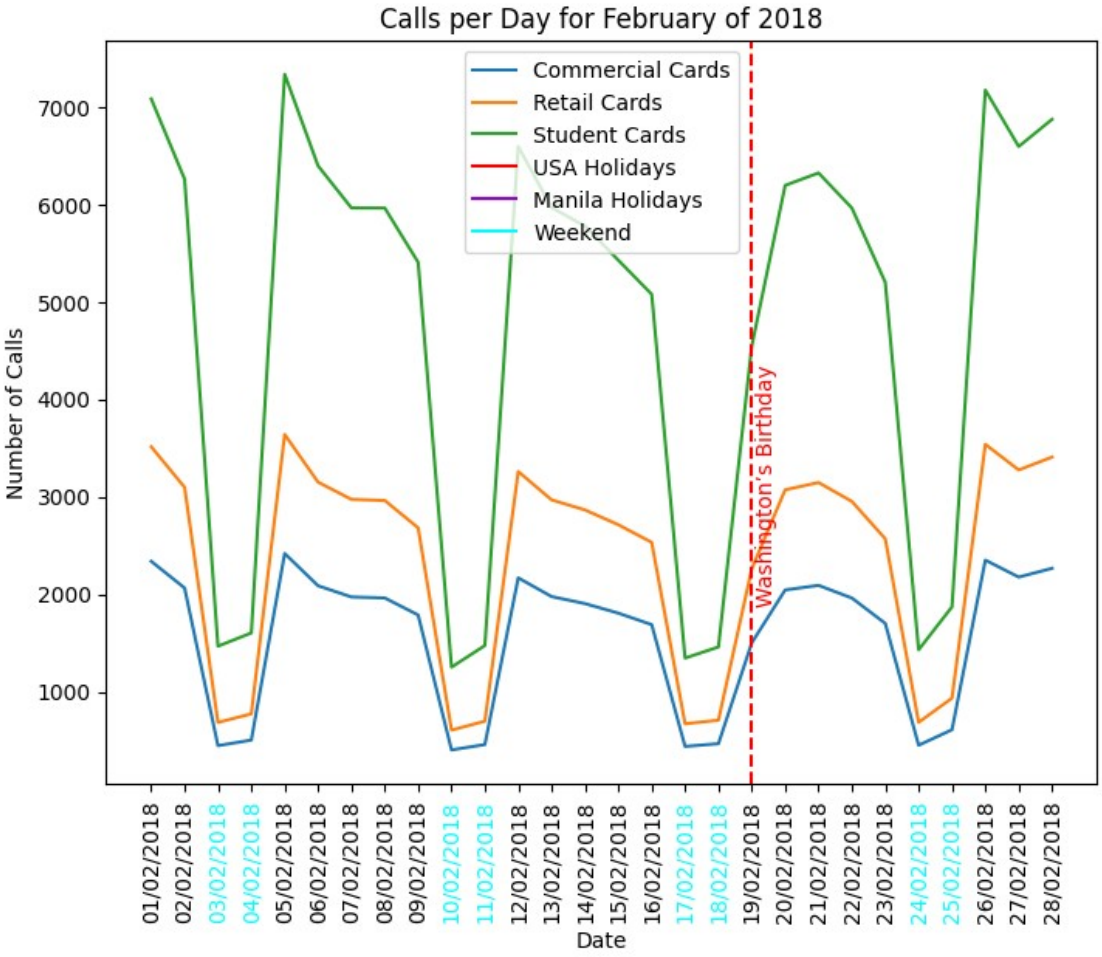

Calls per Day for November
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
2000
4000
6000
8000
Veterans Day
Thanksgiving Day
Black Friday
Bonifacio
01/11/2017
02/11/2017
03/11/2017
04/11/2017
05/11/2017
06/11/2017
07/11/2017
08/11/2017
09/11/2017
10/11/2017
11/11/2017
12/11/2017
13/11/2017
14/11/2017
15/11/2017
16/11/2017
17/11/2017
18/11/2017
19/11/2017
20/11/2017
21/11/2017
22/11/2017
23/11/2017
24/11/2017
25/11/2017
26/11/2017
27/11/2017
28/11/2017
29/11/2017
30/11/2017
Date

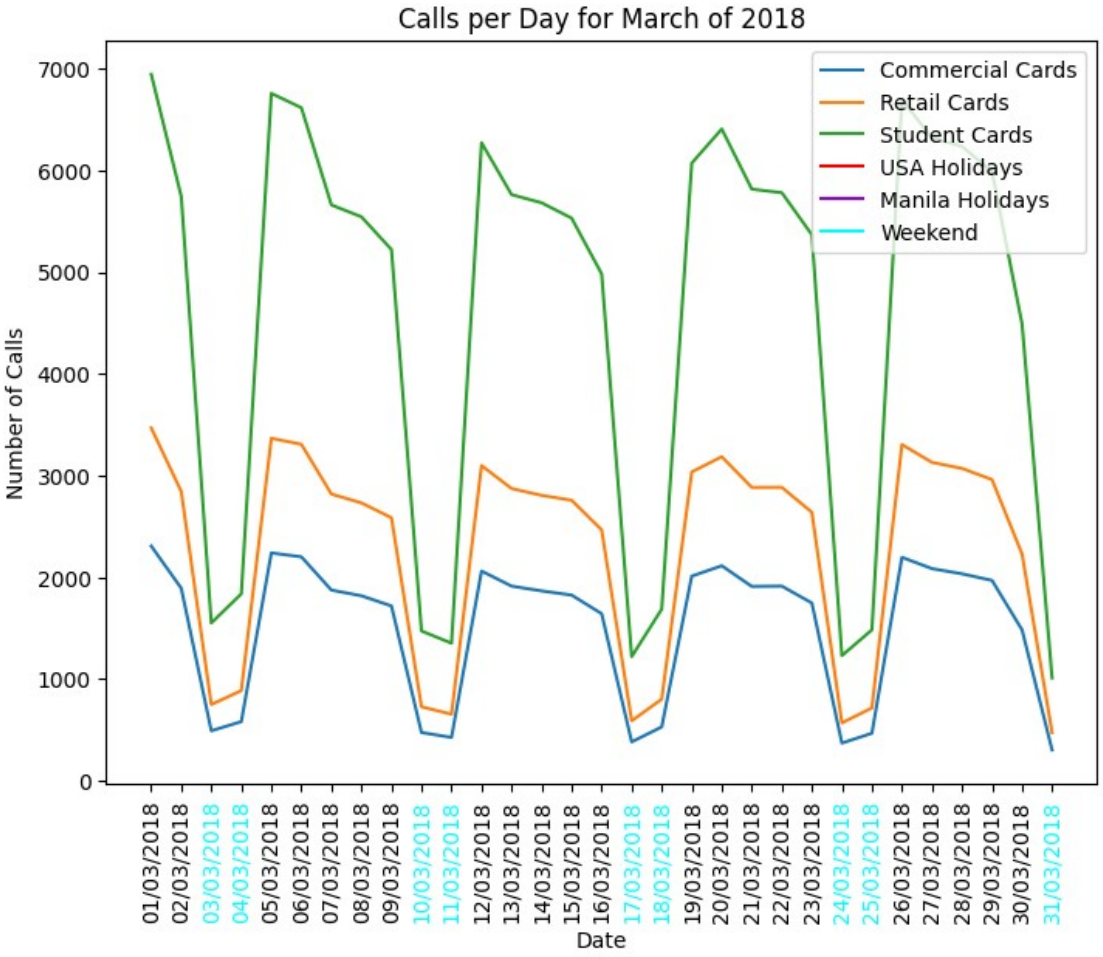


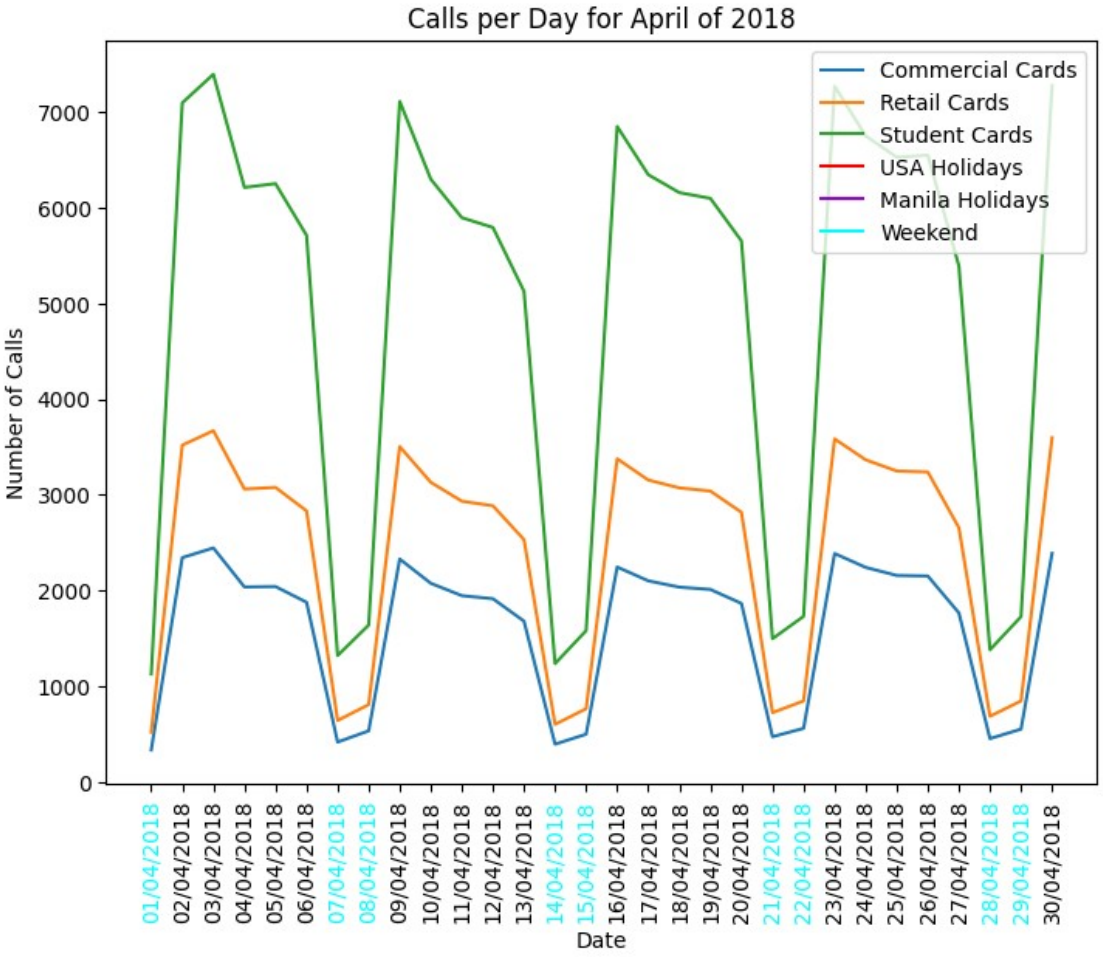

```
# 2018-2021 data for each month
# Low number of calls on weekends
# Low number of calls on Big Holidays
years = df_not17_sorted.year.unique()
for y in years:
    holidays_yearly = holidays[[True if dt.year == y else False for dt
in holidays.date]]
    m_holidays_yearly = m_holidays[[True if dt.year == y else False
for dt in m_holidays.date]]
    df_not17_sorted_yearly = df_not17_sorted[df_not17_sorted.year ==
y]
    months = df_not17_sorted_yearly.month.unique()
    for m in months:
        x = range((df_not17_sorted_yearly.month == m).sum())
        cc_values =
df_not17_sorted_yearly[df_not17_sorted_yearly.month == m].cc
        rc_values =
df_not17_sorted_yearly[df_not17_sorted_yearly.month == m].rc
        sc_values =
df_not17_sorted_yearly[df_not17_sorted_yearly.month == m].sc
        month_holidays = holidays_yearly[[True if dt.month == m else
```

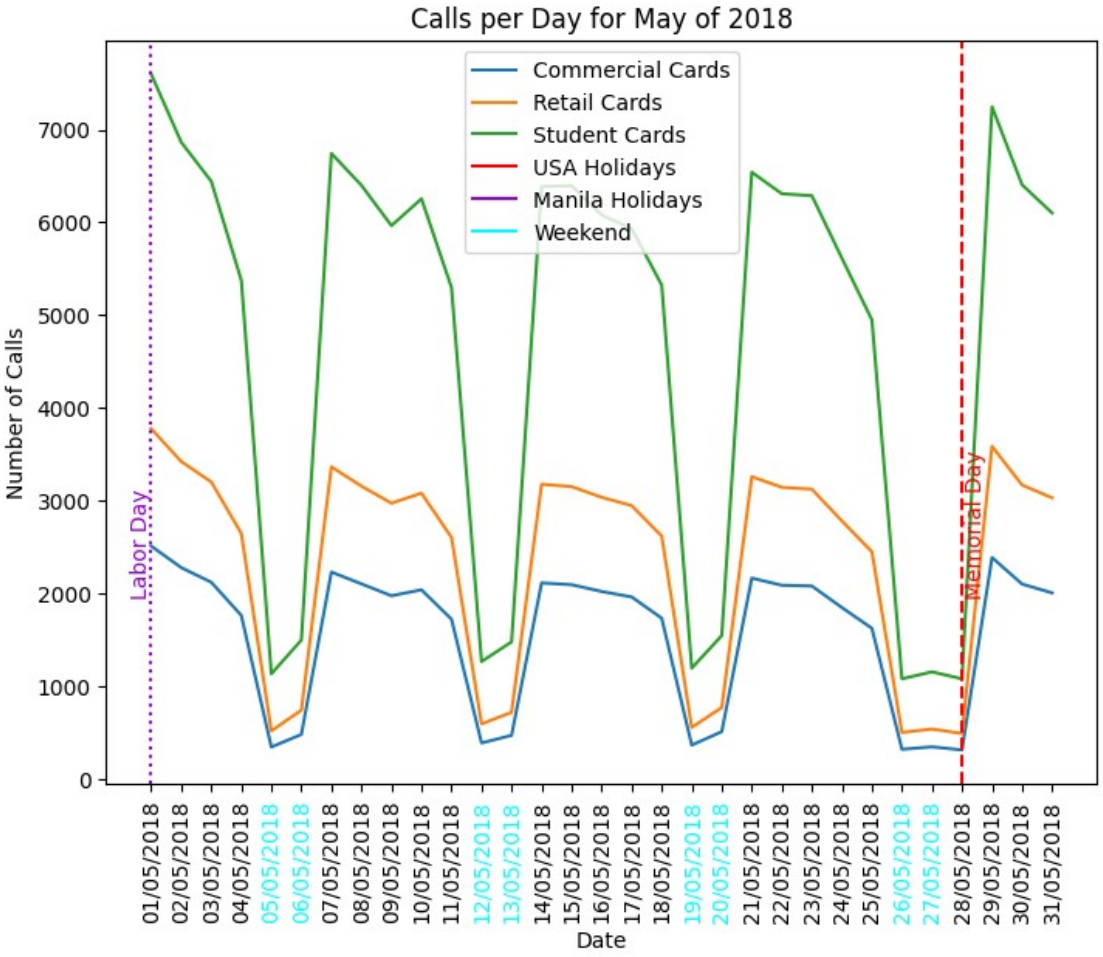
```
False for dt in holidays_yearly.date]]
        monhol_values = [(dt.date.day-1, dt['name']) for i, dt in
month_holidays.iterrows()]
        m_month_holidays = m_holidays_yearly[[True if dt.month == m
else False for dt in m_holidays_yearly.date]]
        m_monhol_values = [(dt.date.day-1, dt['name']) for i, dt in
m_month_holidays.iterrows()]
        fig = plt.figure()
        ax = fig.add_axes([1,1,1,1])
        ax.plot(x, cc_values, label='Commercial Cards')
        ax.plot(x, rc_values, label='Retail Cards')
        ax.plot(x, sc_values, label='Student Cards')
        ax.plot([], [], label='USA Holidays', c='red')
        ax.plot([], [], label='Manila Holidays', c='darkviolet')
        ax.plot([], [], label='Weekend', c='cyan')
        for mhd, mhn in monhol_values:
            if mhn == 'Thanksgiving Day':
                ax.axvline(mhd+1, c='red', ls='--')
                ax.text(mhd+1.1, ax.get_ylim()[1]//4, "Black Friday",
rotation=90, c='red')
            if mhn == 'Christmas Day':
                ax.axvline(mhd-1, c='red', ls='--')
                ax.text(mhd-0.9, ax.get_ylim()[1]//4, "Christman Eve",
rotation=90, c='red')
            ax.axvline(mhd, c='red', ls='--')
            ax.text(mhd+0.1, ax.get_ylim()[1]//4, mhn, rotation=90,
c='red')
            for mhd, mhn in m_monhol_values:
                ax.axvline(mhd, c='darkviolet', ls=':')
                ax.text(mhd-0.7, ax.get_ylim()[1]//4, mhn,
rotation=90, c='darkviolet')
        ax.set_xticks(x,
df_not17_sorted_yearly[df_not17_sorted_yearly.month == m].date,
rotation=90)
        for i, dt in
enumerate(df_not17_sorted_yearly[df_not17_sorted_yearly.month ==
m].date):
            if arrow.get(dt, "DD/MM/YYYY").format("ddd") in ["Sat",
"Sun"]:
                ax.get_xticklabels()[i].set_color('cyan')
        ax.set_xlabel("Date")
        ax.set_ylabel("Number of Calls")
        ax.set_title(f"Calls per Day for {arrow.get(str(m),
'M').format('MMMM')} of {y}")
        ax.legend()
        plt.show()
```

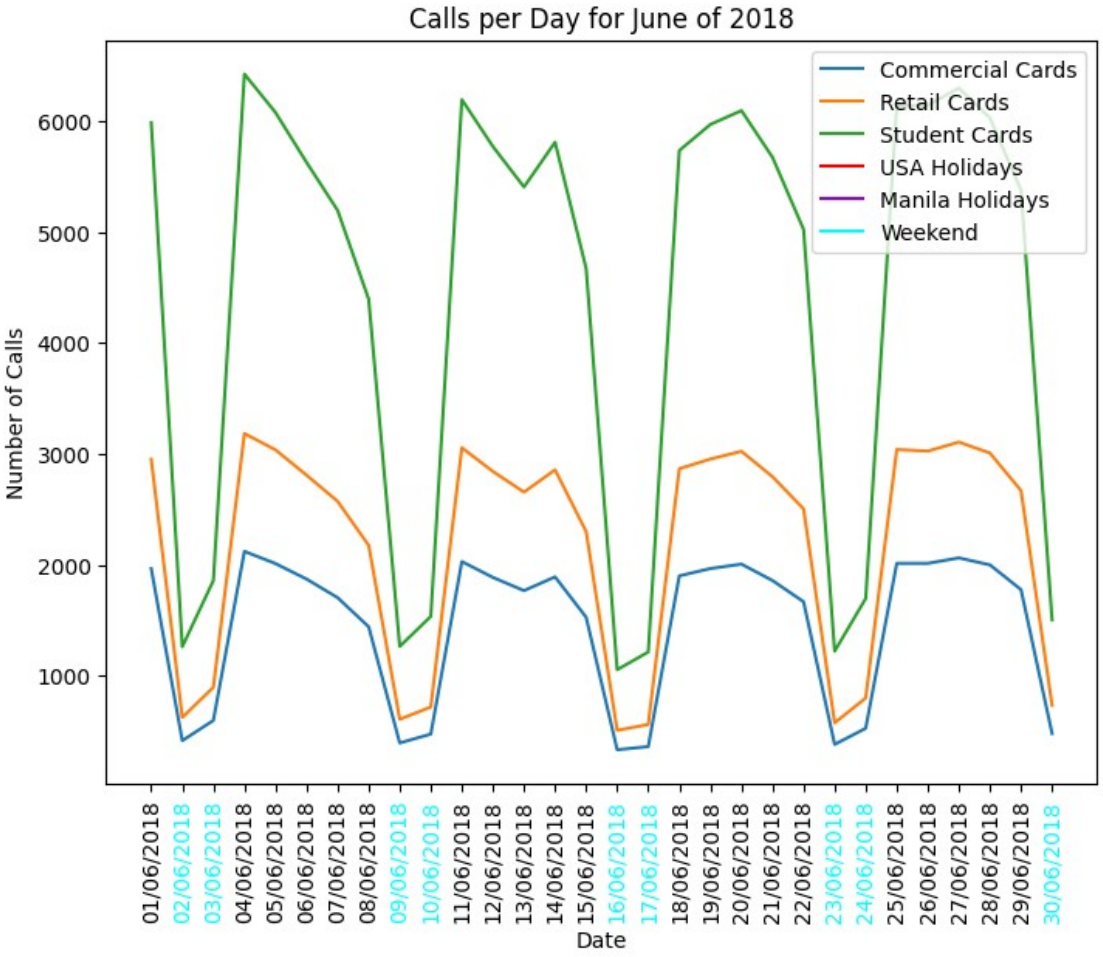

Calls per Day for January of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
New Year
New Year's Day
Birthday of Martin Luther King, J
01/01/2018
02/01/2018
03/01/2018
04/01/2018
05/01/2018
06/01/2018
07/01/2018
08/01/2018
09/01/2018
10/01/2018
11/01/2018
12/01/2018
13/01/2018
14/01/2018
15/01/2018
16/01/2018
17/01/2018
18/01/2018
19/01/2018
20/01/2018
21/01/2018
22/01/2018
23/01/2018
24/01/2018
25/01/2018
26/01/2018
27/01/2018
28/01/2018
29/01/2018
30/01/2018
31/01/2018
Date

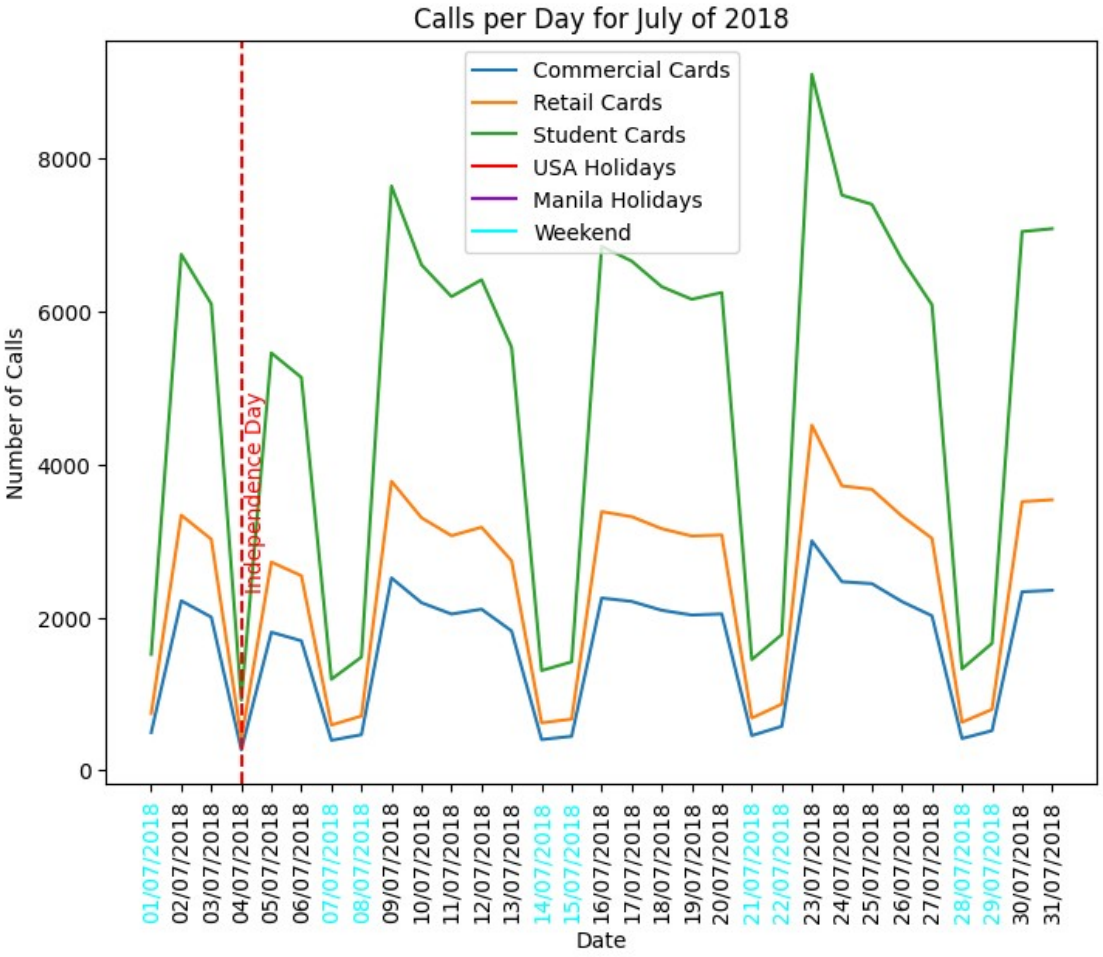

Calls per Day for February of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
7000
6000
5000
4000
3000
2000
1000
Washington's Birthday
01/02/2018
02/02/2018
03/02/2018
04/02/2018
05/02/2018
06/02/2018
07/02/2018
08/02/2018
09/02/2018
10/02/2018
11/02/2018
12/02/2018
13/02/2018
14/02/2018
15/02/2018
16/02/2018
17/02/2018
18/02/2018
19/02/2018
20/02/2018
21/02/2018
22/02/2018
23/02/2018
24/02/2018
25/02/2018
26/02/2018
27/02/2018
28/02/2018
Date

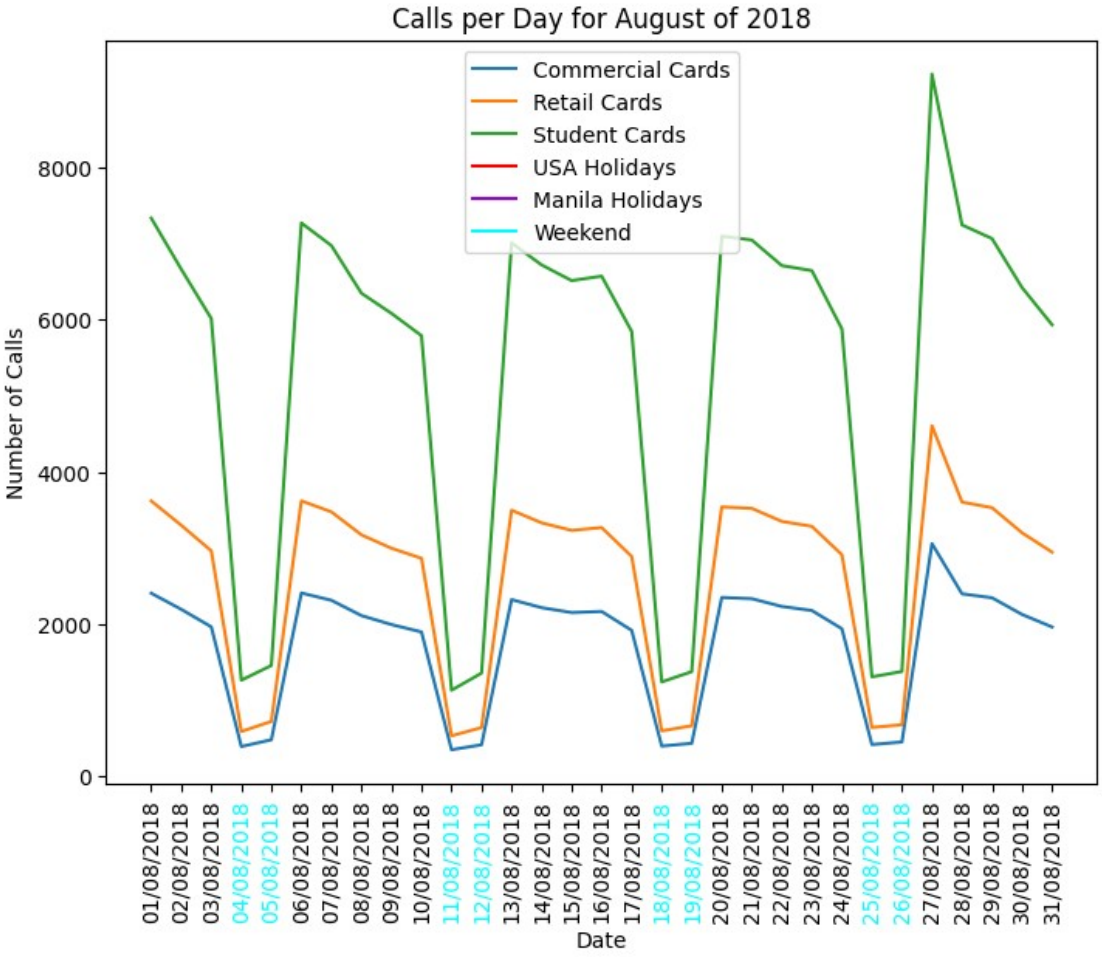

Calls per Day for March of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/03/2018
02/03/2018
03/03/2018
04/03/2018
05/03/2018
06/03/2018
07/03/2018
08/03/2018
09/03/2018
10/03/2018
11/03/2018
12/03/2018
13/03/2018
14/03/2018
15/03/2018
16/03/2018
17/03/2018
18/03/2018
19/03/2018
20/03/2018
21/03/2018
22/03/2018
23/03/2018
24/03/2018
25/03/2018
26/03/2018
27/03/2018
28/03/2018
29/03/2018
30/03/2018
31/03/2018
Date

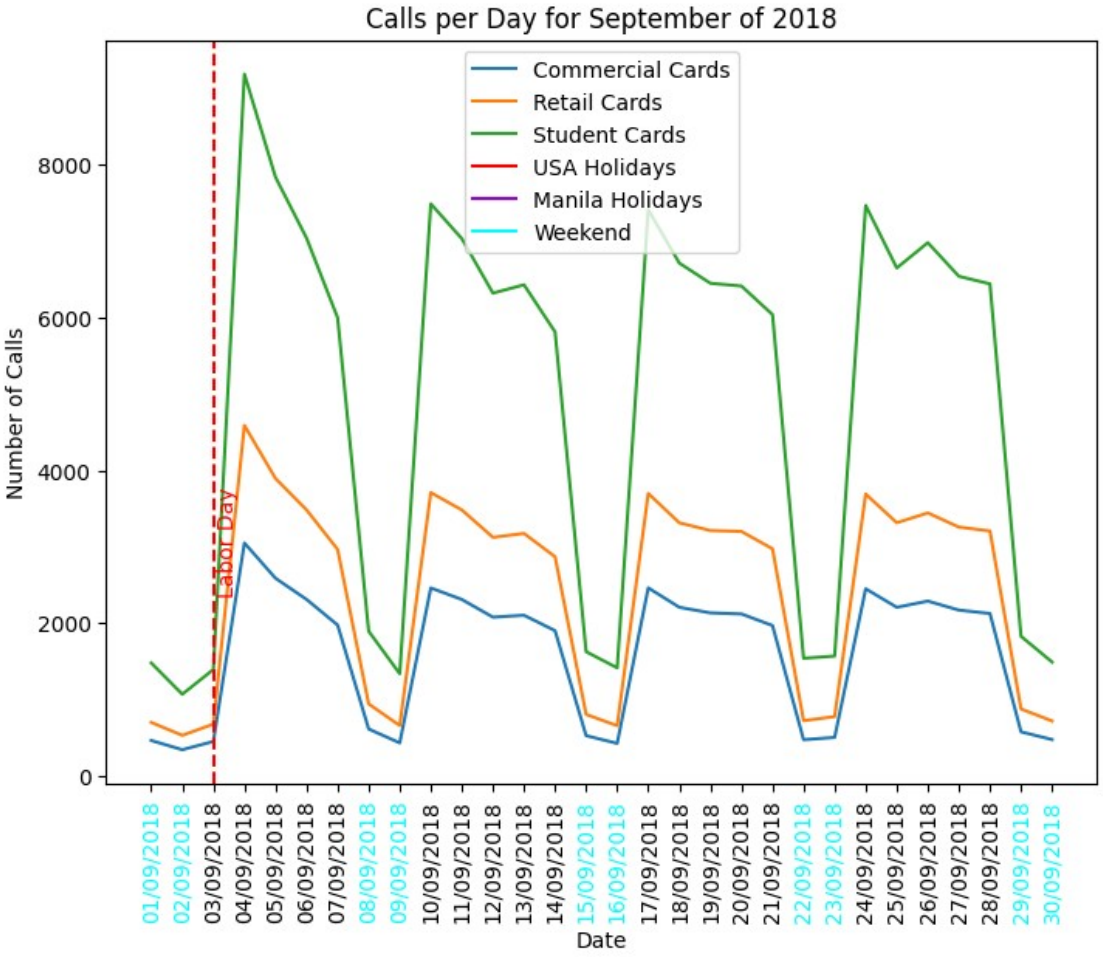

Calls per Day for April of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/04/2018
02/04/2018
03/04/2018
04/04/2018
05/04/2018
06/04/2018
07/04/2018
08/04/2018
09/04/2018
10/04/2018
11/04/2018
12/04/2018
13/04/2018
14/04/2018
15/04/2018
16/04/2018
17/04/2018
18/04/2018
19/04/2018
20/04/2018
21/04/2018
22/04/2018
23/04/2018
24/04/2018
25/04/2018
26/04/2018
27/04/2018
28/04/2018
29/04/2018
30/04/2018
Date

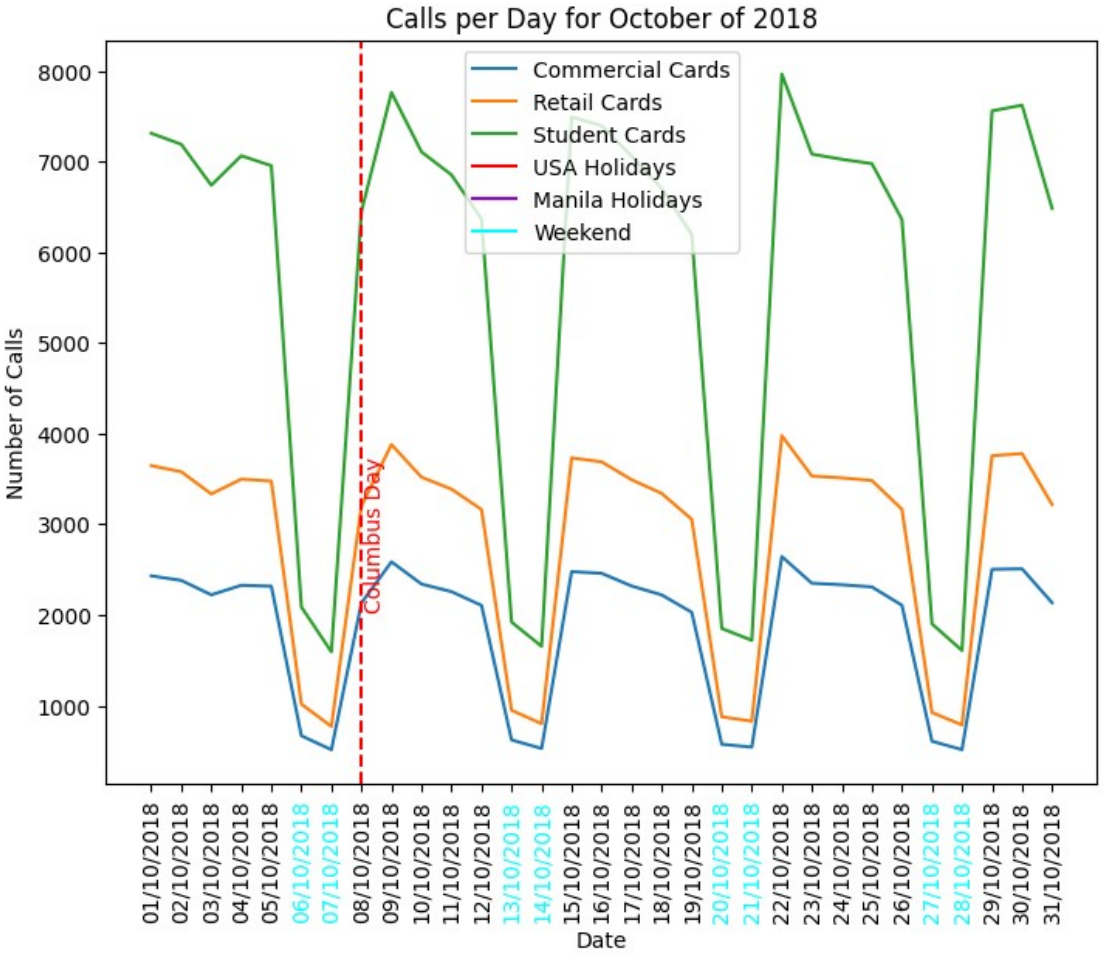

Calls per Day for May of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
Labor Day
Memorial Day
01/05/2018
02/05/2018
03/05/2018
04/05/2018
05/05/2018
06/05/2018
07/05/2018
08/05/2018
09/05/2018
10/05/2018
11/05/2018
12/05/2018
13/05/2018
14/05/2018
15/05/2018
16/05/2018
17/05/2018
18/05/2018
19/05/2018
20/05/2018
21/05/2018
22/05/2018
23/05/2018
24/05/2018
25/05/2018
26/05/2018
27/05/2018
28/05/2018
29/05/2018
30/05/2018
31/05/2018
Date

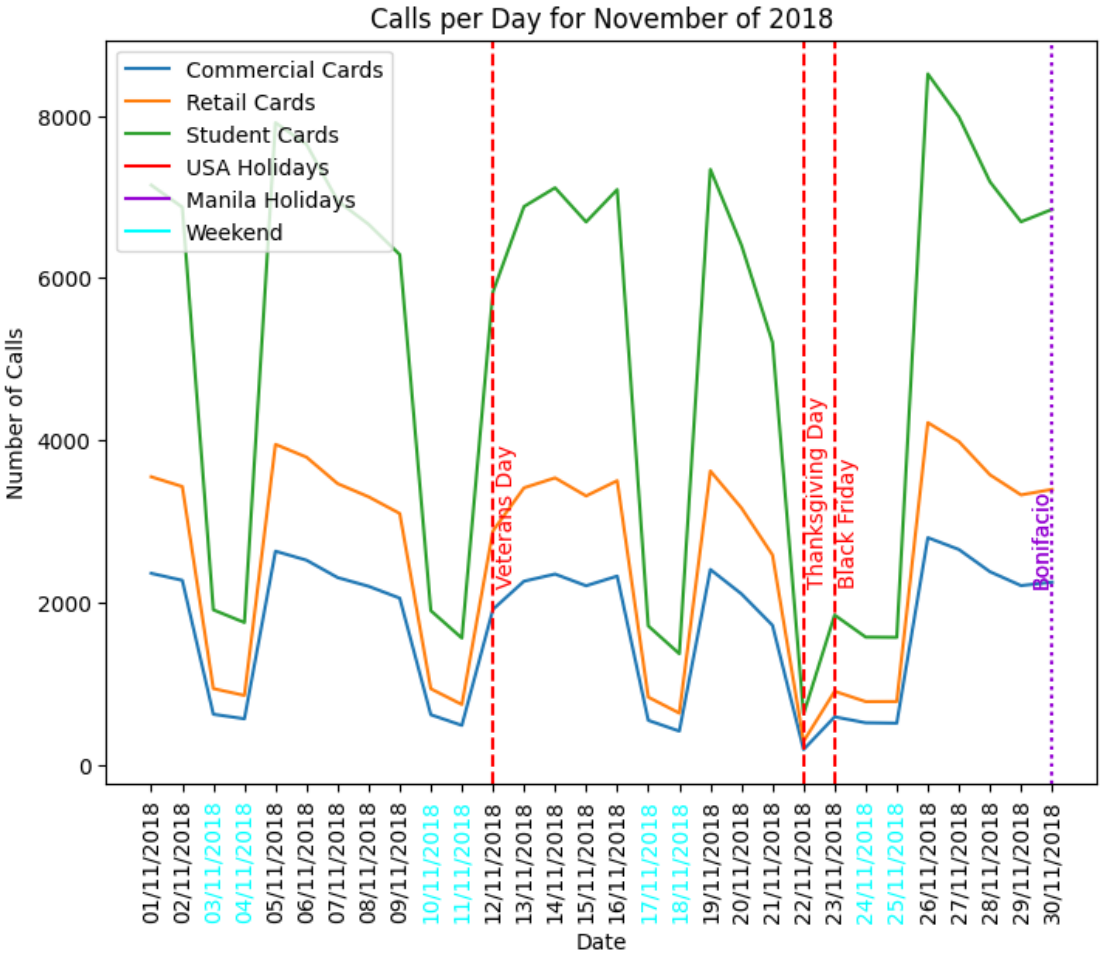

Calls per Day for June of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
1000
2000
3000
4000
5000
6000
01/06/2018
02/06/2018
03/06/2018
04/06/2018
05/06/2018
06/06/2018
07/06/2018
08/06/2018
09/06/2018
10/06/2018
11/06/2018
12/06/2018
13/06/2018
14/06/2018
15/06/2018
16/06/2018
17/06/2018
18/06/2018
19/06/2018
20/06/2018
21/06/2018
22/06/2018
23/06/2018
24/06/2018
25/06/2018
26/06/2018
27/06/2018
28/06/2018
29/06/2018
30/06/2018
Date

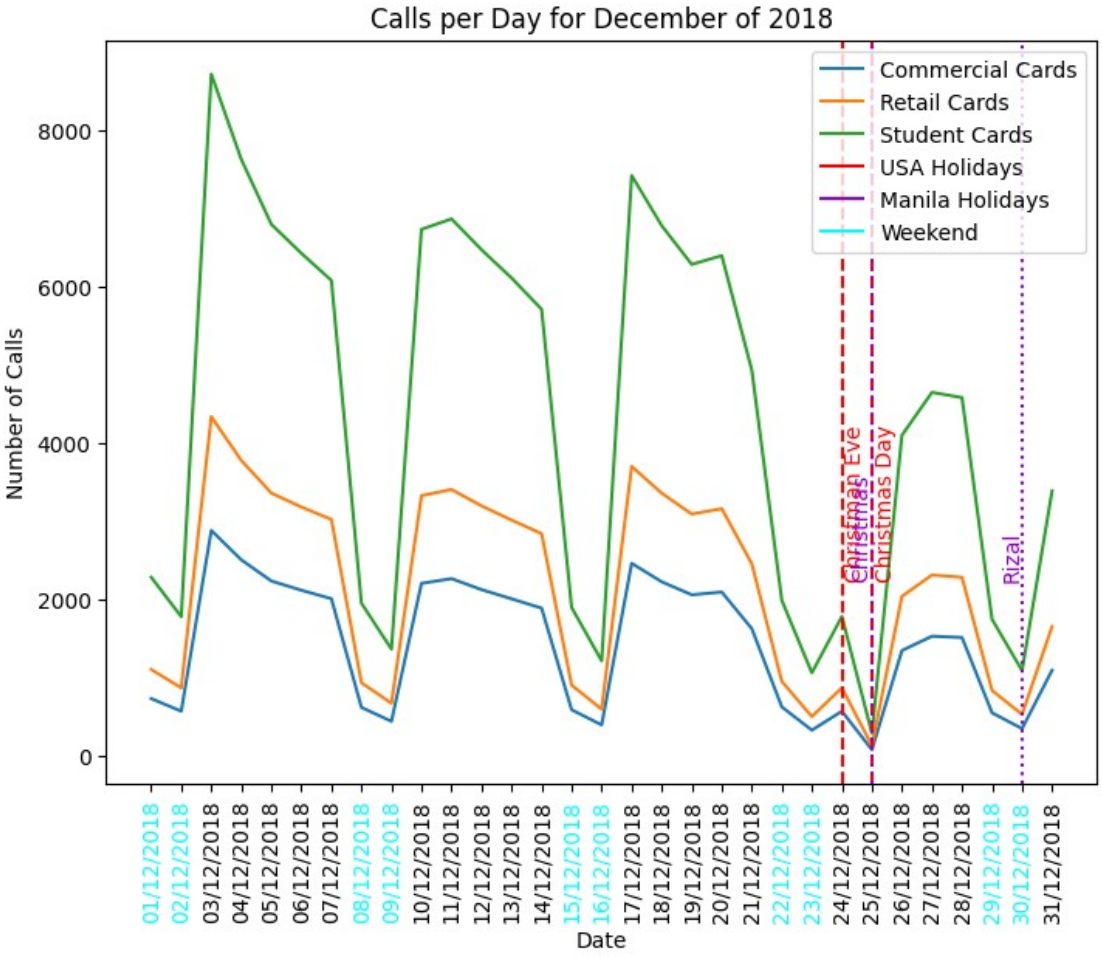

Calls per Day for July of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Independence Day
Number of Calls
0
2000
4000
6000
8000
01/07/2018
02/07/2018
03/07/2018
04/07/2018
05/07/2018
06/07/2018
07/07/2018
08/07/2018
09/07/2018
10/07/2018
11/07/2018
12/07/2018
13/07/2018
14/07/2018
15/07/2018
16/07/2018
17/07/2018
18/07/2018
19/07/2018
20/07/2018
21/07/2018
22/07/2018
23/07/2018
24/07/2018
25/07/2018
26/07/2018
27/07/2018
28/07/2018
29/07/2018
30/07/2018
31/07/2018
Date

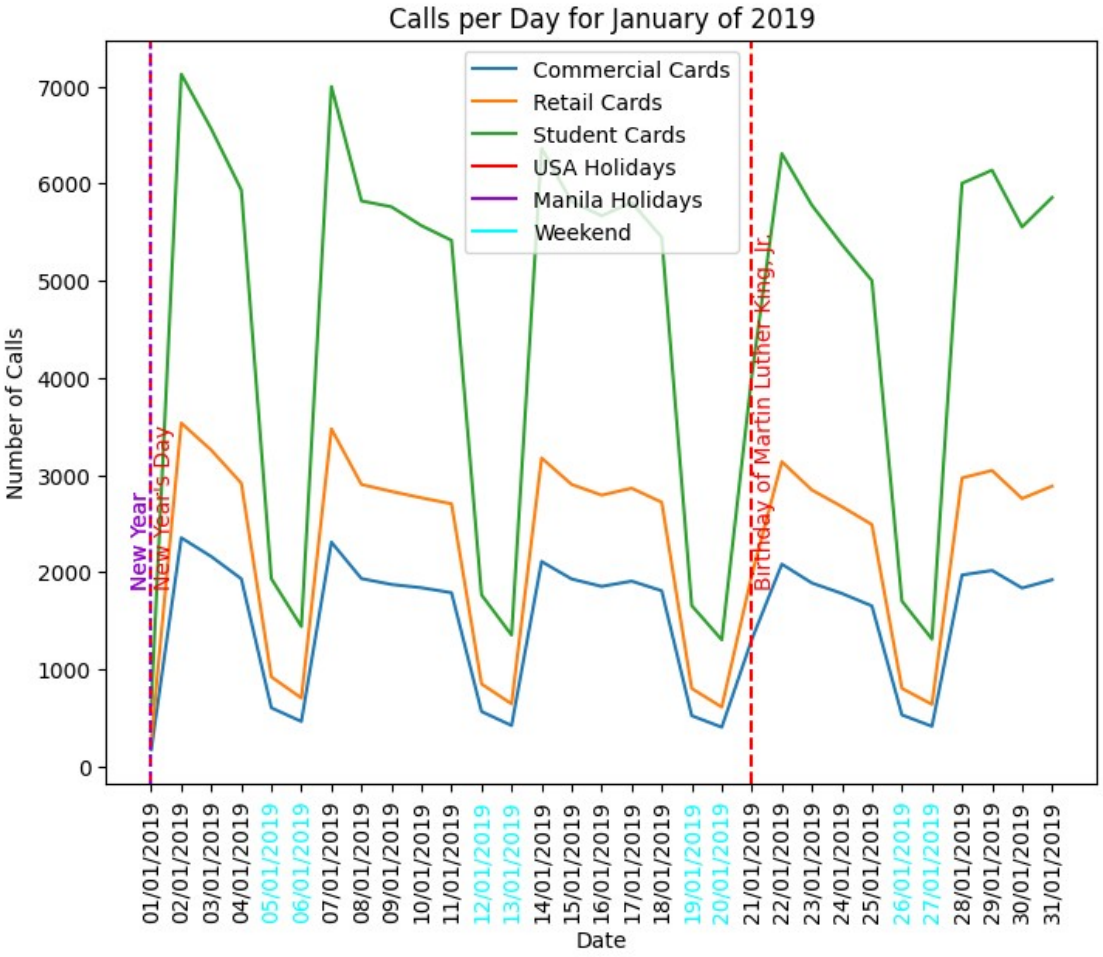

Calls per Day for August of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
2000
4000
6000
8000
01/08/2018
02/08/2018
03/08/2018
04/08/2018
05/08/2018
06/08/2018
07/08/2018
08/08/2018
09/08/2018
10/08/2018
11/08/2018
12/08/2018
13/08/2018
14/08/2018
15/08/2018
16/08/2018
17/08/2018
18/08/2018
19/08/2018
20/08/2018
21/08/2018
22/08/2018
23/08/2018
24/08/2018
25/08/2018
26/08/2018
27/08/2018
28/08/2018
29/08/2018
30/08/2018
31/08/2018
Date

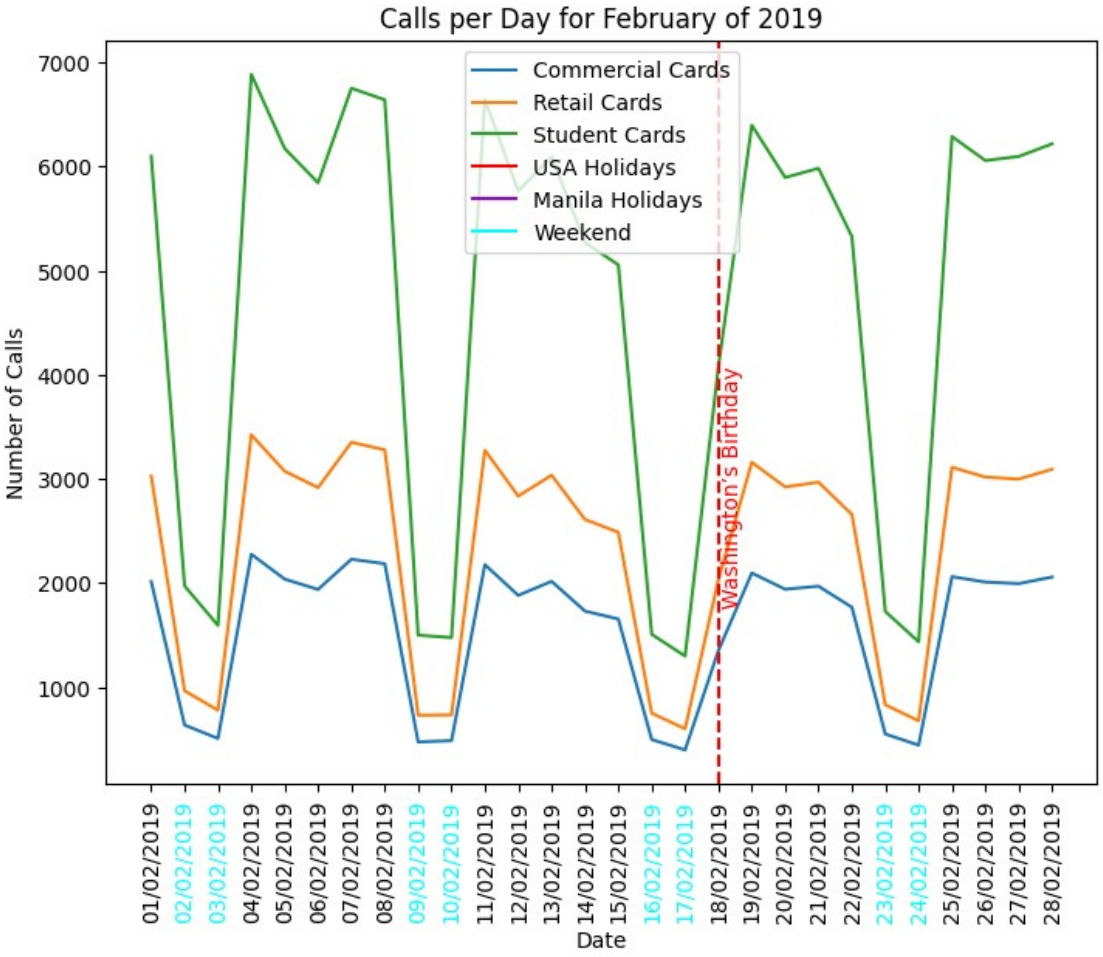

Calls per Day for September of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
2000
4000
6000
8000
Labor Day
01/09/2018
02/09/2018
03/09/2018
04/09/2018
05/09/2018
06/09/2018
07/09/2018
08/09/2018
09/09/2018
10/09/2018
11/09/2018
12/09/2018
13/09/2018
14/09/2018
15/09/2018
16/09/2018
17/09/2018
18/09/2018
19/09/2018
20/09/2018
21/09/2018
22/09/2018
23/09/2018
24/09/2018
25/09/2018
26/09/2018
27/09/2018
28/09/2018
29/09/2018
30/09/2018
Date

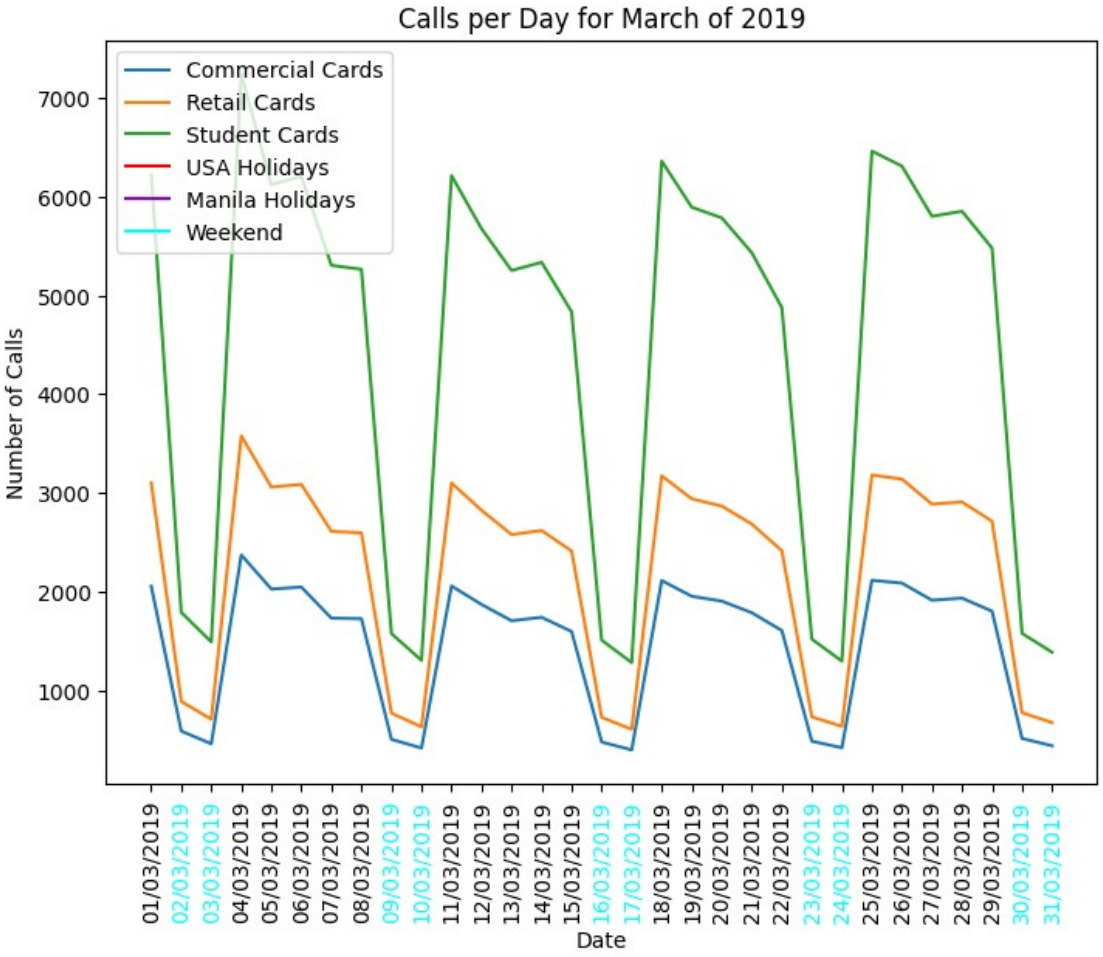

Calls per Day for October of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Columbus Day
Number of Calls
8000
7000
6000
5000
4000
3000
2000
1000
01/10/2018
02/10/2018
03/10/2018
04/10/2018
05/10/2018
06/10/2018
07/10/2018
08/10/2018
09/10/2018
10/10/2018
11/10/2018
12/10/2018
13/10/2018
14/10/2018
15/10/2018
16/10/2018
17/10/2018
18/10/2018
19/10/2018
20/10/2018
21/10/2018
22/10/2018
23/10/2018
24/10/2018
25/10/2018
26/10/2018
27/10/2018
28/10/2018
29/10/2018
30/10/2018
31/10/2018
Date

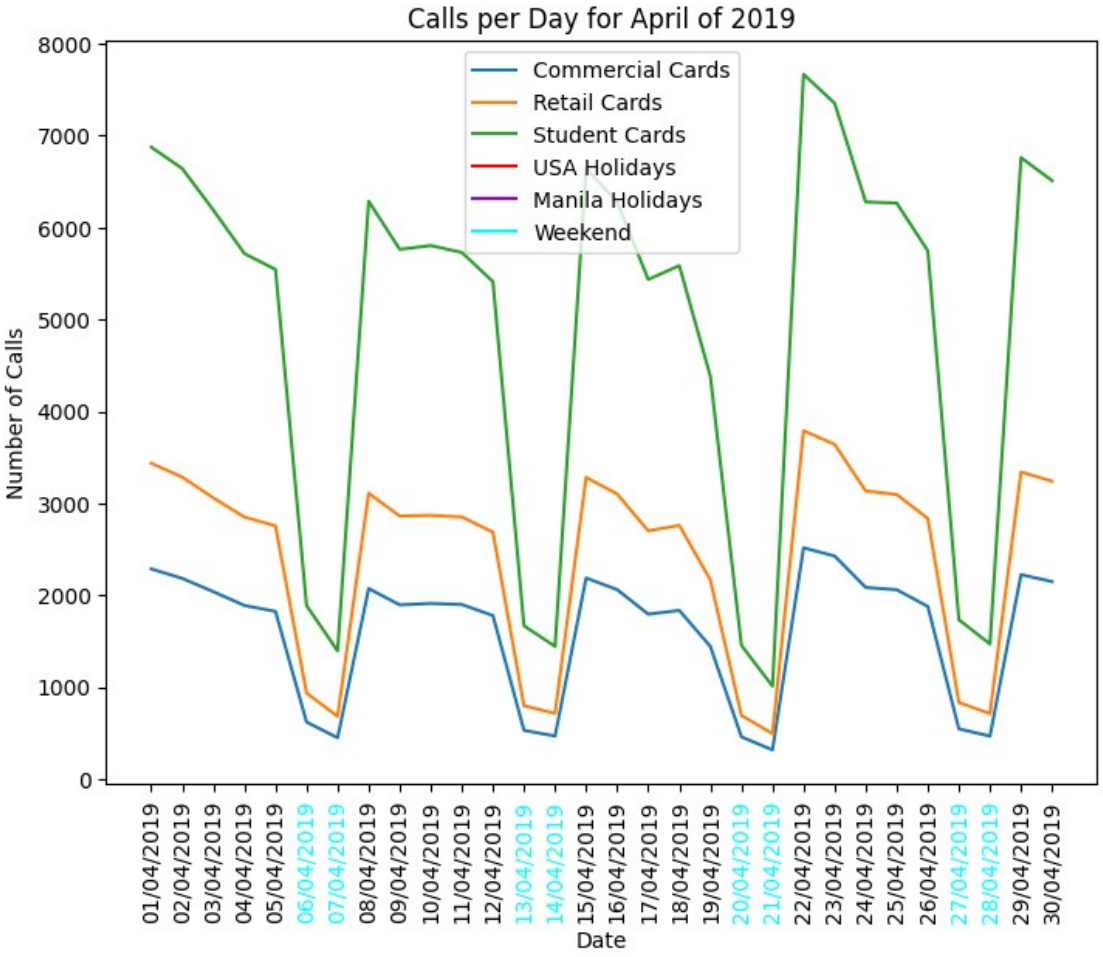

Calls per Day for November of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
2000
4000
6000
8000
Veterans Day
Thanksgiving Day
Black Friday
Bonifacio
01/11/2018
02/11/2018
03/11/2018
04/11/2018
05/11/2018
06/11/2018
07/11/2018
08/11/2018
09/11/2018
10/11/2018
11/11/2018
12/11/2018
13/11/2018
14/11/2018
15/11/2018
16/11/2018
17/11/2018
18/11/2018
19/11/2018
20/11/2018
21/11/2018
22/11/2018
23/11/2018
24/11/2018
25/11/2018
26/11/2018
27/11/2018
28/11/2018
29/11/2018
30/11/2018
Date

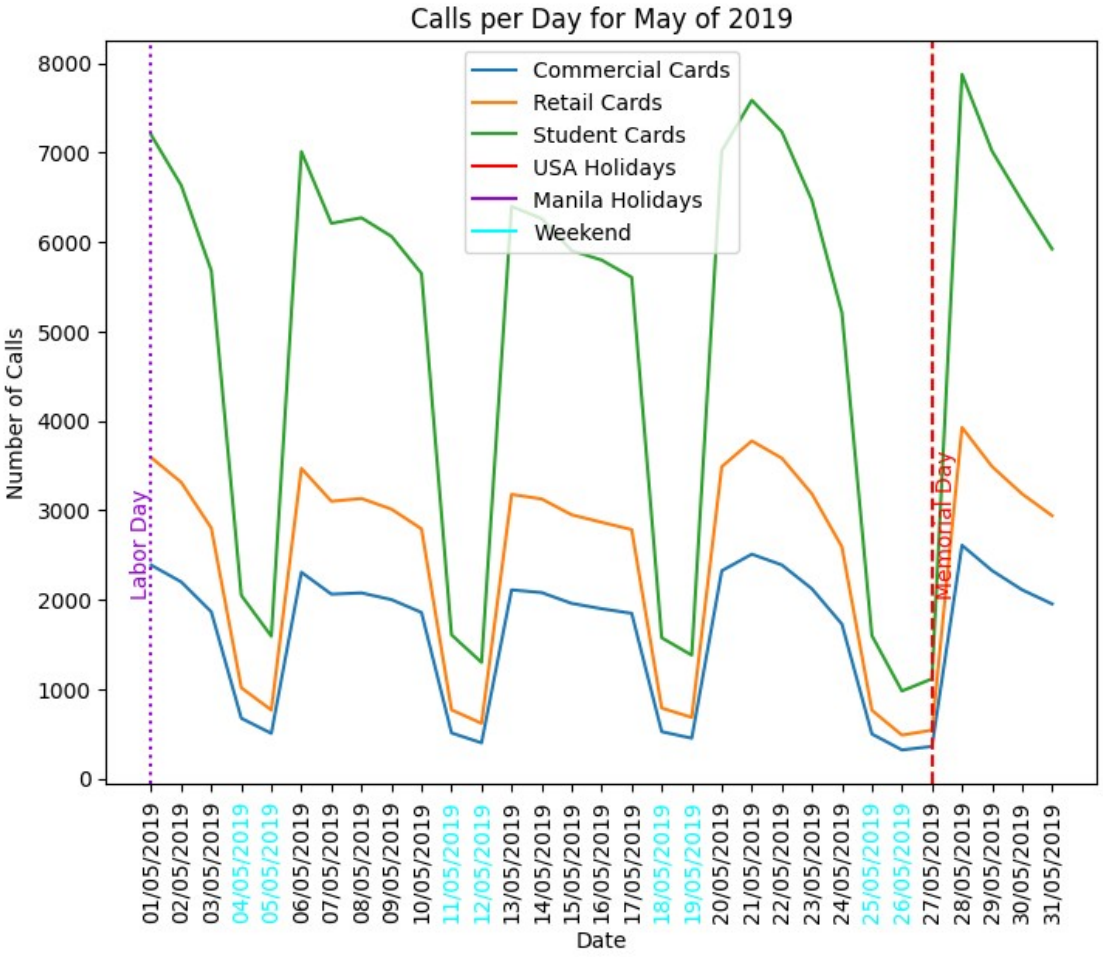

Calls per Day for December of 2018
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
2000
4000
6000
8000
Christmas Eve
Christmas Day
Rizal
01/12/2018
02/12/2018
03/12/2018
04/12/2018
05/12/2018
06/12/2018
07/12/2018
08/12/2018
09/12/2018
10/12/2018
11/12/2018
12/12/2018
13/12/2018
14/12/2018
15/12/2018
16/12/2018
17/12/2018
18/12/2018
19/12/2018
20/12/2018
21/12/2018
22/12/2018
23/12/2018
24/12/2018
25/12/2018
26/12/2018
27/12/2018
28/12/2018
29/12/2018
30/12/2018
31/12/2018
Date

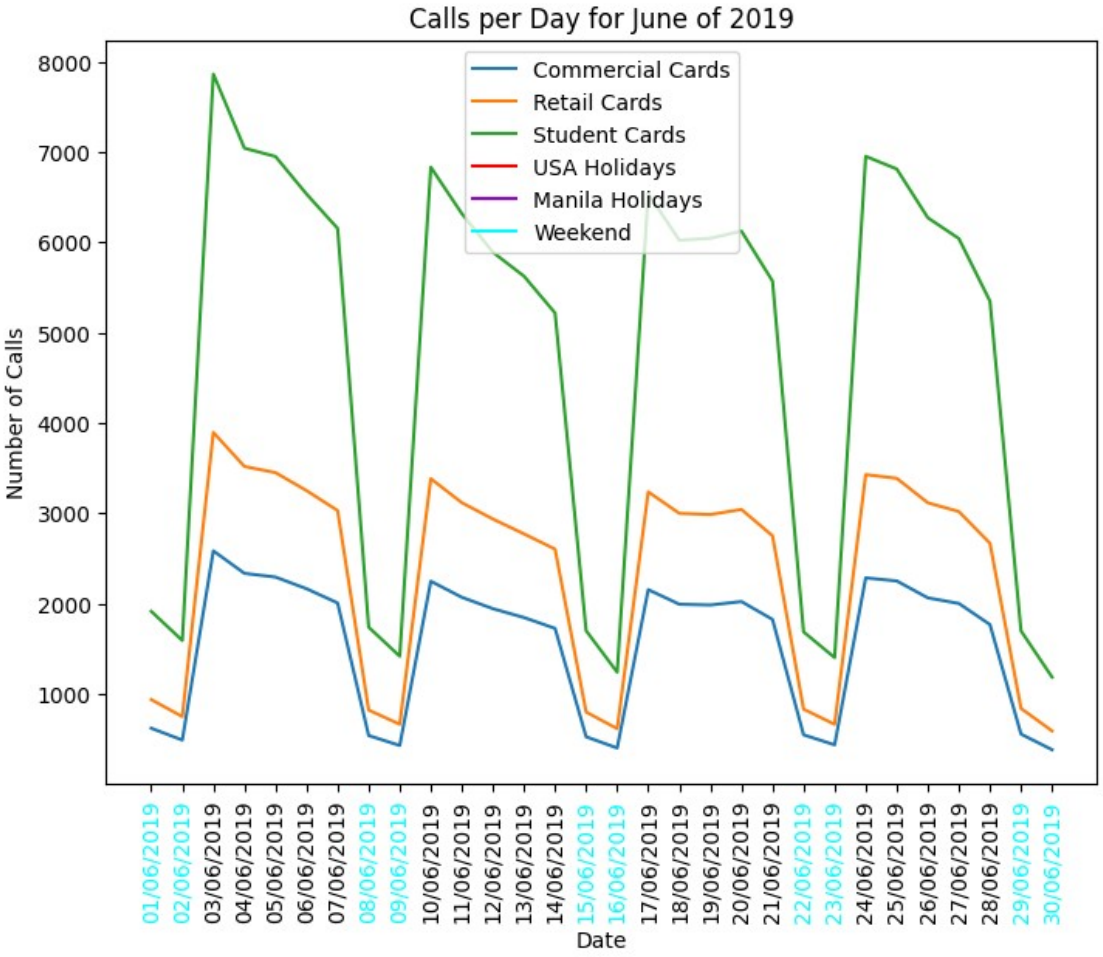

Calls per Day for January of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
New Year
New Year's Day
Birthday of Martin Luther King, Jr.
01/01/2019
02/01/2019
03/01/2019
04/01/2019
05/01/2019
06/01/2019
07/01/2019
08/01/2019
09/01/2019
10/01/2019
11/01/2019
12/01/2019
13/01/2019
14/01/2019
15/01/2019
16/01/2019
17/01/2019
18/01/2019
19/01/2019
20/01/2019
21/01/2019
22/01/2019
23/01/2019
24/01/2019
25/01/2019
26/01/2019
27/01/2019
28/01/2019
29/01/2019
30/01/2019
31/01/2019
Date

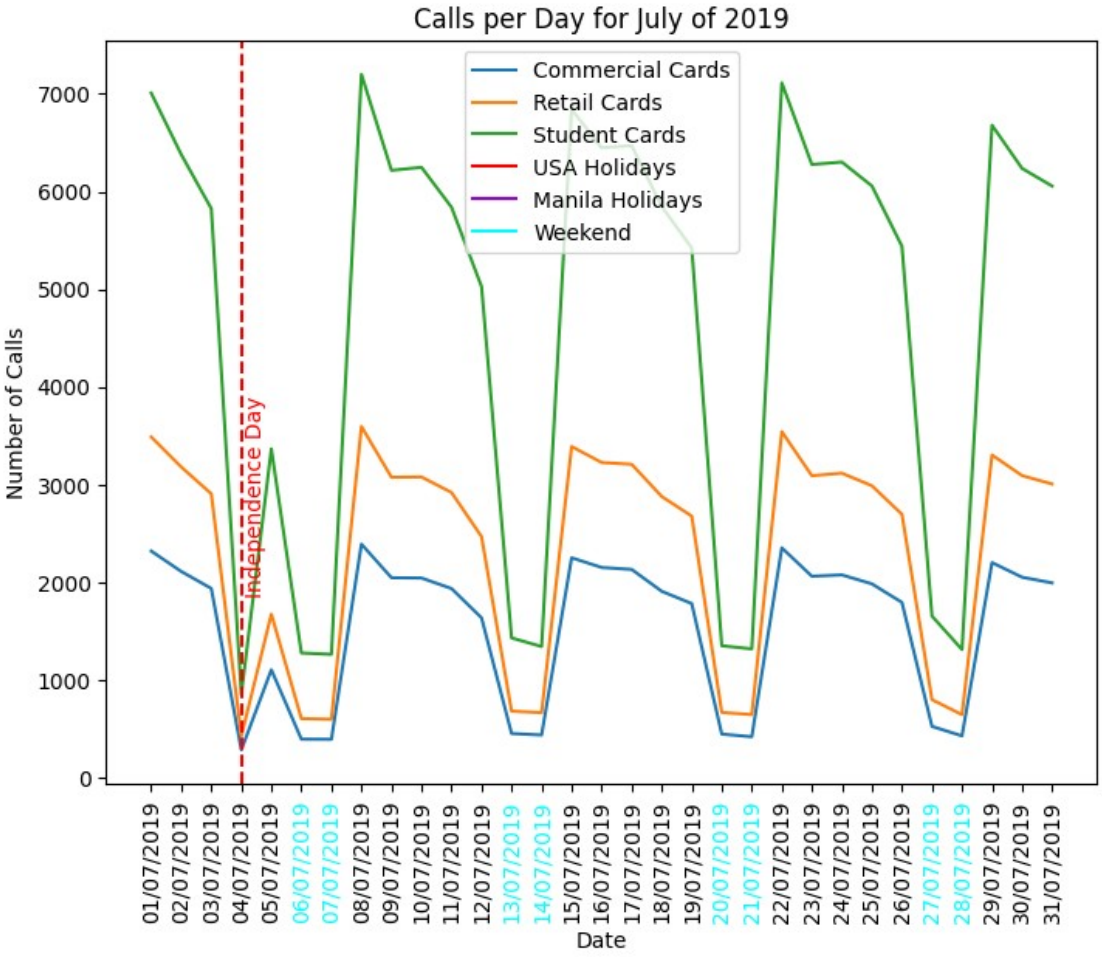

Calls per Day for February of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Washington's Birthday
Number of Calls
7000
6000
5000
4000
3000
2000
1000
01/02/2019
02/02/2019
03/02/2019
04/02/2019
05/02/2019
06/02/2019
07/02/2019
08/02/2019
09/02/2019
10/02/2019
11/02/2019
12/02/2019
13/02/2019
14/02/2019
15/02/2019
16/02/2019
17/02/2019
18/02/2019
19/02/2019
20/02/2019
21/02/2019
22/02/2019
23/02/2019
24/02/2019
25/02/2019
26/02/2019
27/02/2019
28/02/2019
Date

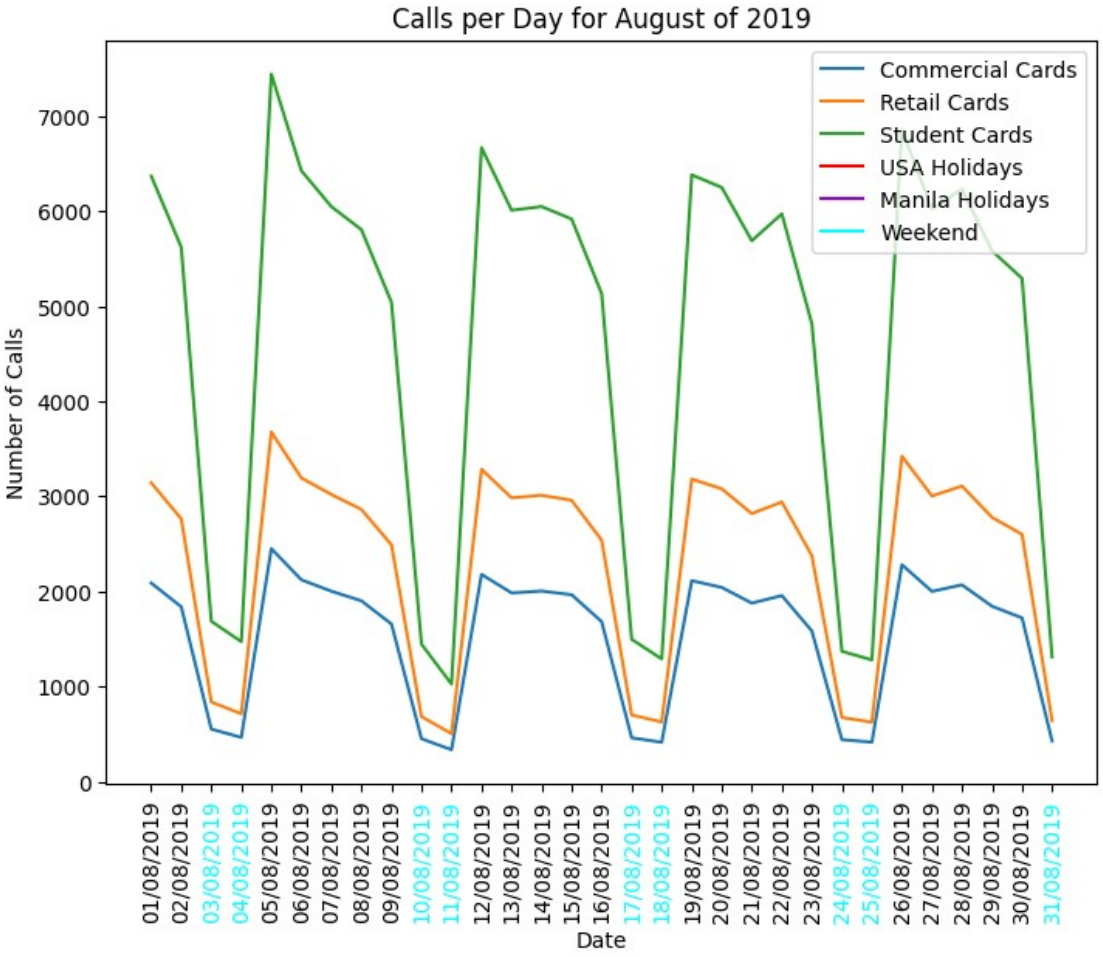

Calls per Day for March of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
7000
6000
5000
4000
3000
2000
1000
01/03/2019
02/03/2019
03/03/2019
04/03/2019
05/03/2019
06/03/2019
07/03/2019
08/03/2019
09/03/2019
10/03/2019
11/03/2019
12/03/2019
13/03/2019
14/03/2019
15/03/2019
16/03/2019
17/03/2019
18/03/2019
19/03/2019
20/03/2019
21/03/2019
22/03/2019
23/03/2019
24/03/2019
25/03/2019
26/03/2019
27/03/2019
28/03/2019
29/03/2019
30/03/2019
31/03/2019
Date

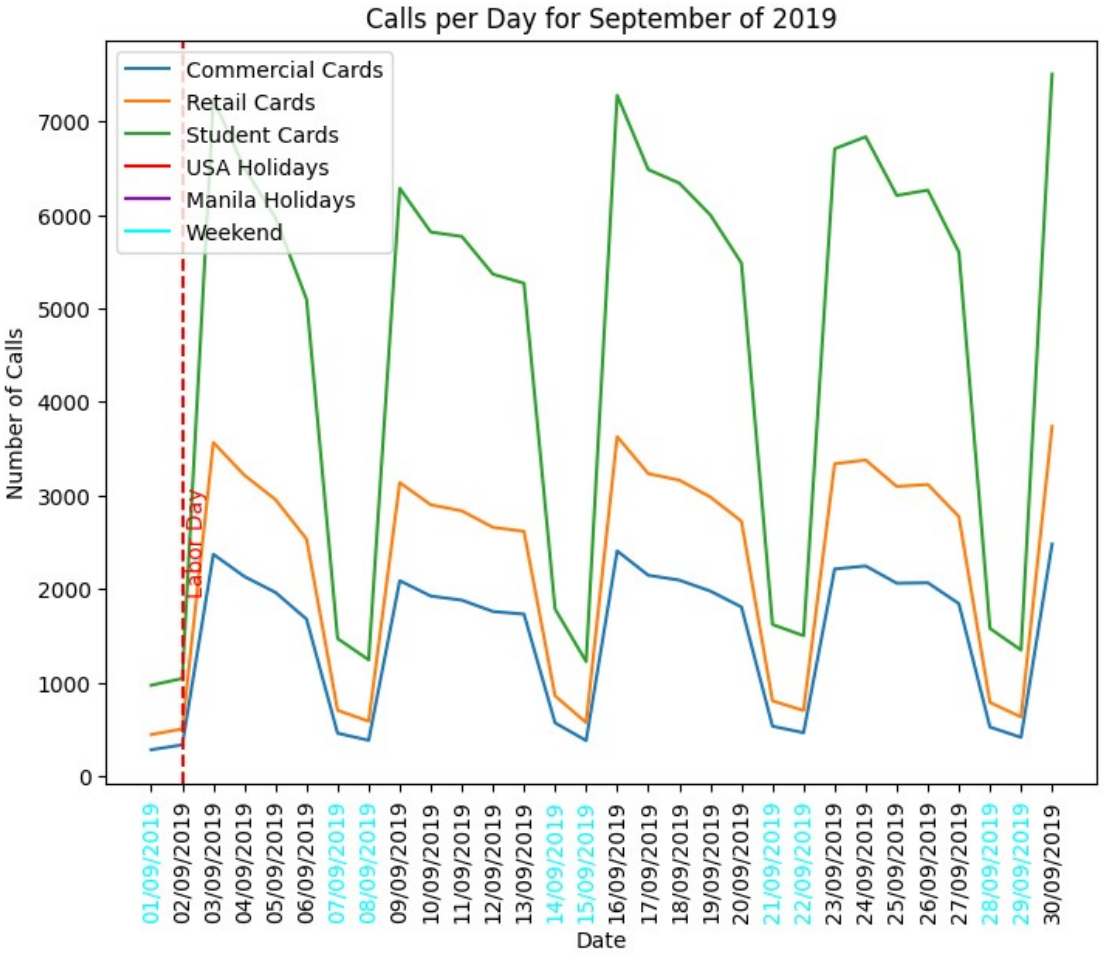

Calls per Day for April of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
8000
7000
6000
5000
4000
3000
2000
1000
0
01/04/2019
02/04/2019
03/04/2019
04/04/2019
05/04/2019
06/04/2019
07/04/2019
08/04/2019
09/04/2019
10/04/2019
11/04/2019
12/04/2019
13/04/2019
14/04/2019
15/04/2019
16/04/2019
17/04/2019
18/04/2019
19/04/2019
20/04/2019
21/04/2019
22/04/2019
23/04/2019
24/04/2019
25/04/2019
26/04/2019
27/04/2019
28/04/2019
29/04/2019
30/04/2019
Date

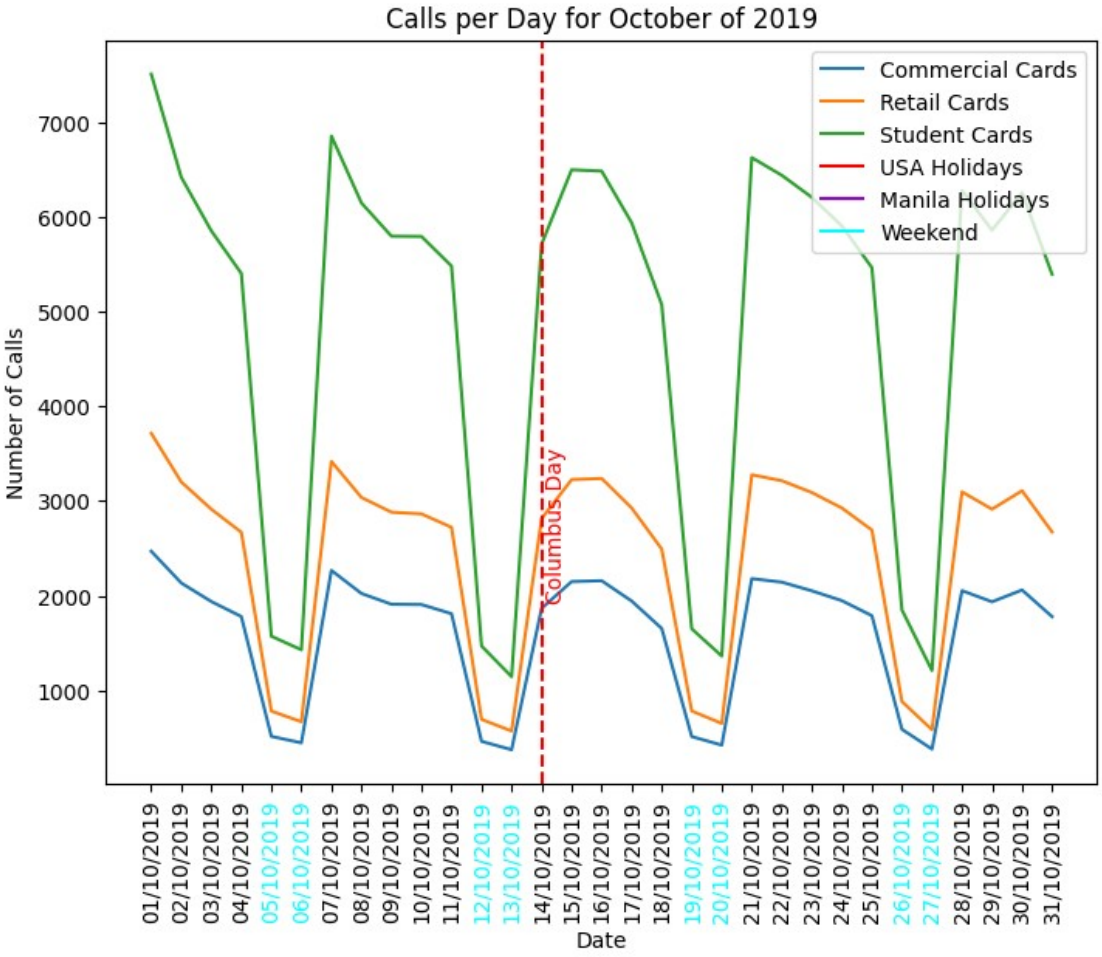

Calls per Day for May of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
Labor Day
Memorial Day
Date

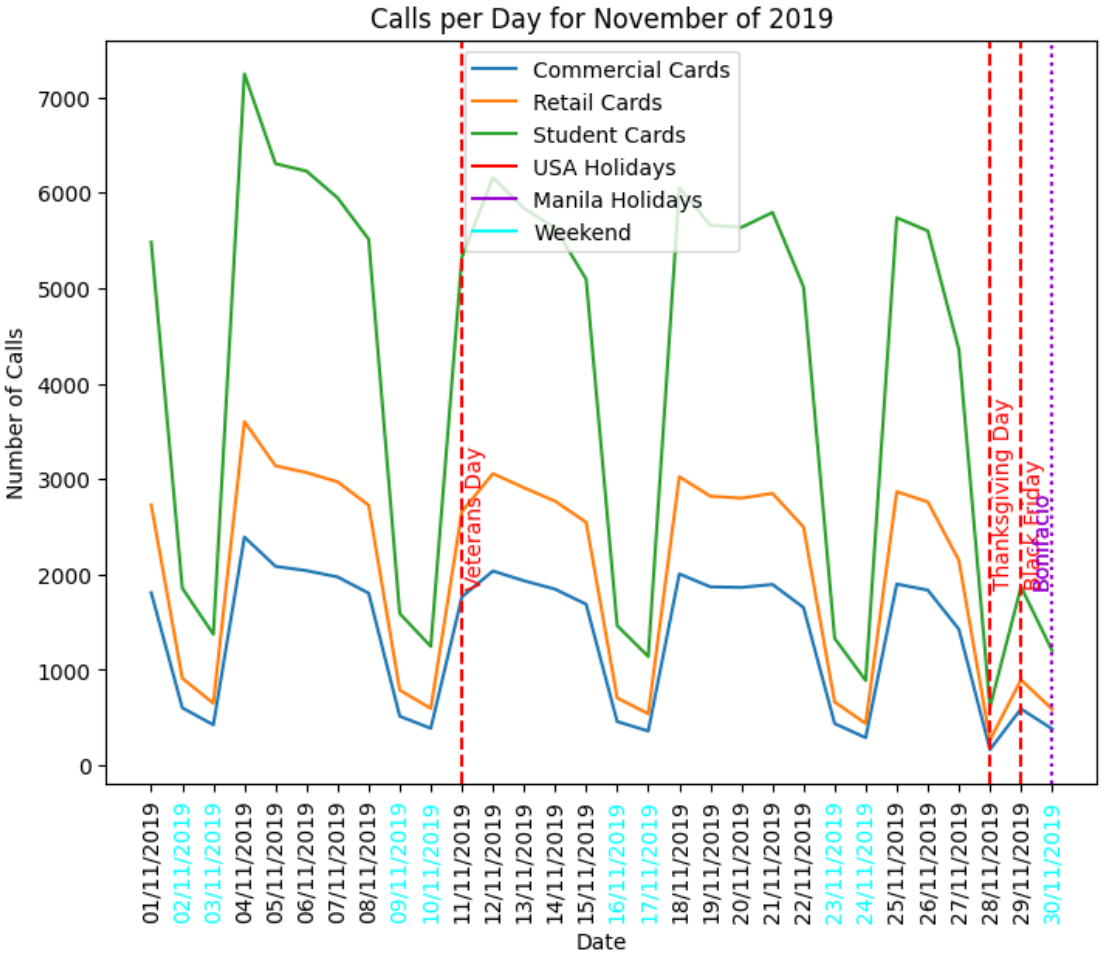

Calls per Day for June of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
8000
7000
6000
5000
4000
3000
2000
1000
01/06/2019
02/06/2019
03/06/2019
04/06/2019
05/06/2019
06/06/2019
07/06/2019
08/06/2019
09/06/2019
10/06/2019
11/06/2019
12/06/2019
13/06/2019
14/06/2019
15/06/2019
16/06/2019
17/06/2019
18/06/2019
19/06/2019
20/06/2019
21/06/2019
22/06/2019
23/06/2019
24/06/2019
25/06/2019
26/06/2019
27/06/2019
28/06/2019
29/06/2019
30/06/2019
Date

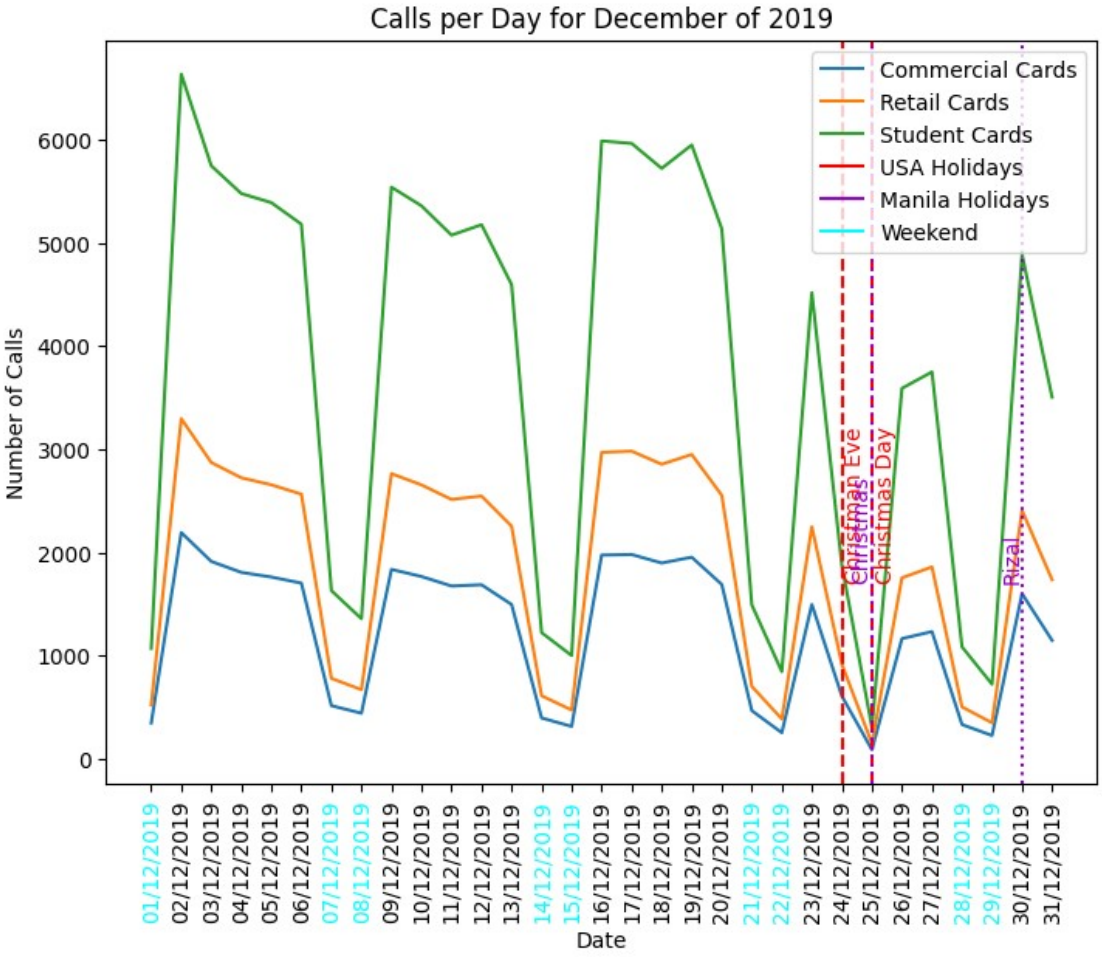

Calls per Day for July of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Independence Day
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/07/2019
02/07/2019
03/07/2019
04/07/2019
05/07/2019
06/07/2019
07/07/2019
08/07/2019
09/07/2019
10/07/2019
11/07/2019
12/07/2019
13/07/2019
14/07/2019
15/07/2019
16/07/2019
17/07/2019
18/07/2019
19/07/2019
20/07/2019
21/07/2019
22/07/2019
23/07/2019
24/07/2019
25/07/2019
26/07/2019
27/07/2019
28/07/2019
29/07/2019
30/07/2019
31/07/2019
Date

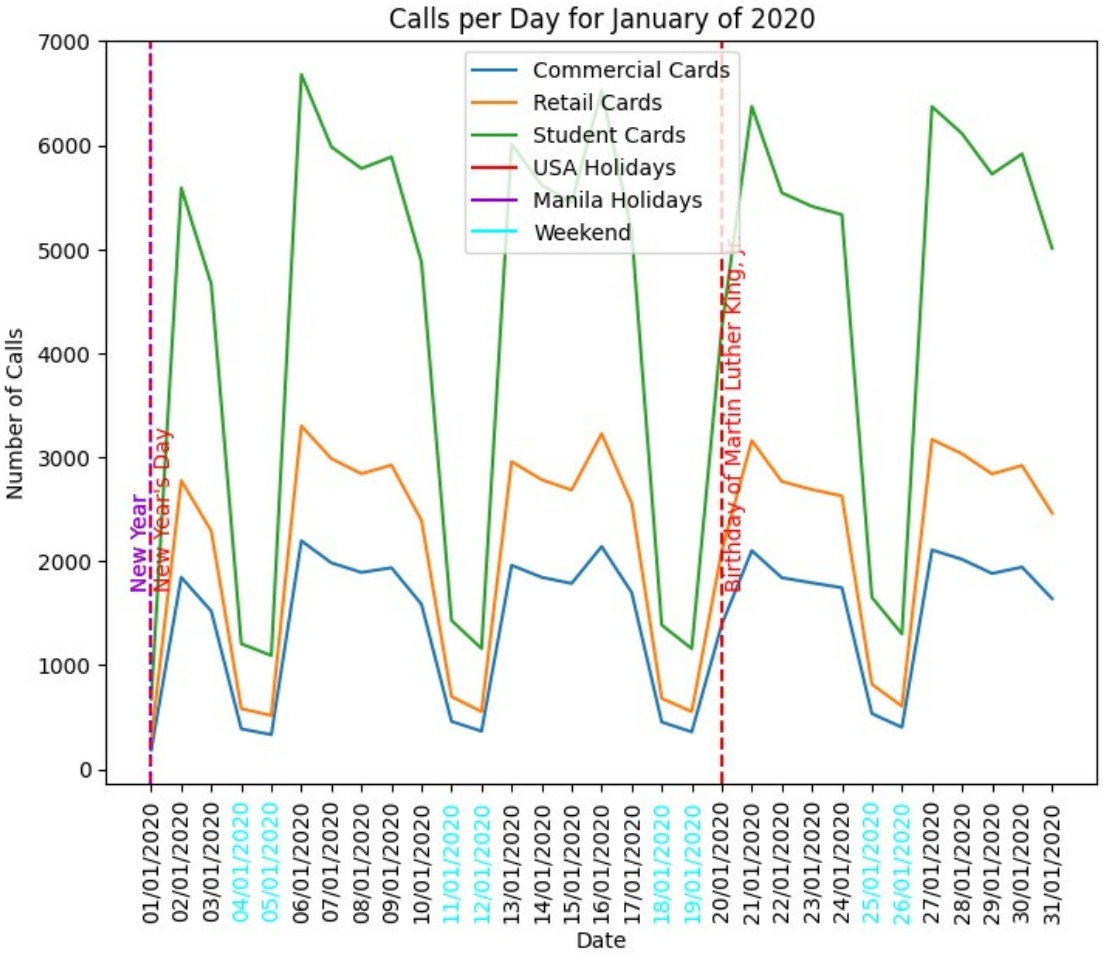

Calls per Day for August of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
01/08/2019
02/08/2019
03/08/2019
04/08/2019
05/08/2019
06/08/2019
07/08/2019
08/08/2019
09/08/2019
10/08/2019
11/08/2019
12/08/2019
13/08/2019
14/08/2019
15/08/2019
16/08/2019
17/08/2019
18/08/2019
19/08/2019
20/08/2019
21/08/2019
22/08/2019
23/08/2019
24/08/2019
25/08/2019
26/08/2019
27/08/2019
28/08/2019
29/08/2019
30/08/2019
31/08/2019
Date

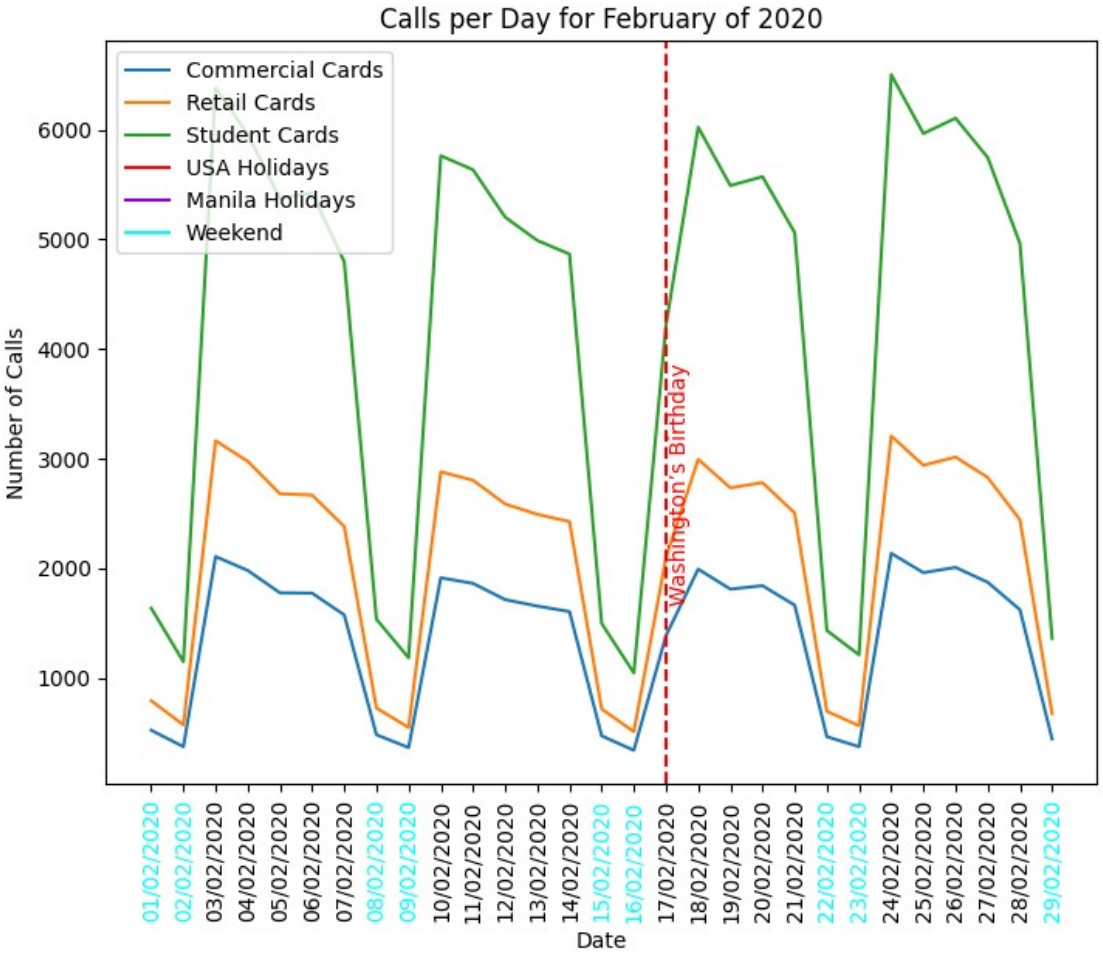

Calls per Day for September of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
Labor Day
Date

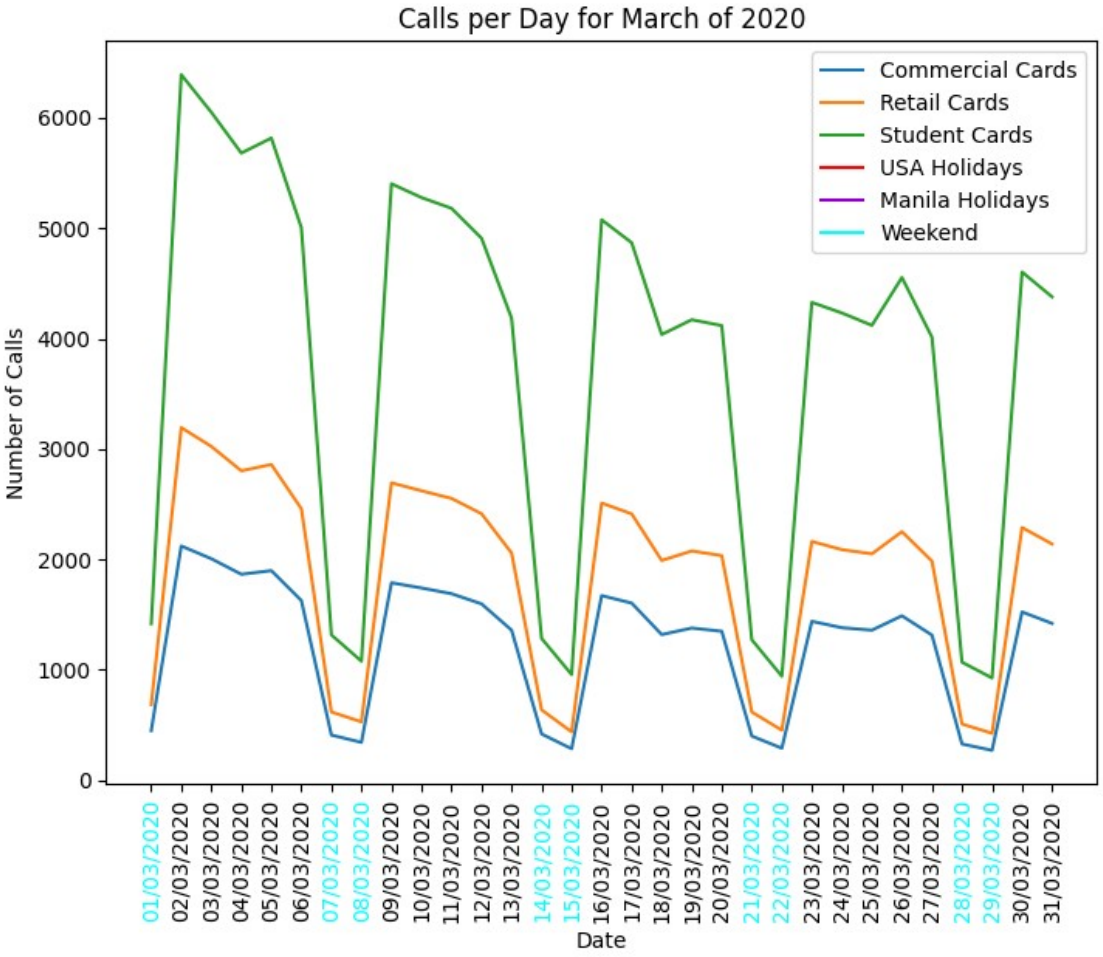

Calls per Day for October of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
7000
6000
5000
4000
3000
2000
1000
Columbus Day
01/10/2019
02/10/2019
03/10/2019
04/10/2019
05/10/2019
06/10/2019
07/10/2019
08/10/2019
09/10/2019
10/10/2019
11/10/2019
12/10/2019
13/10/2019
14/10/2019
15/10/2019
16/10/2019
17/10/2019
18/10/2019
19/10/2019
20/10/2019
21/10/2019
22/10/2019
23/10/2019
24/10/2019
25/10/2019
26/10/2019
27/10/2019
28/10/2019
29/10/2019
30/10/2019
31/10/2019
Date

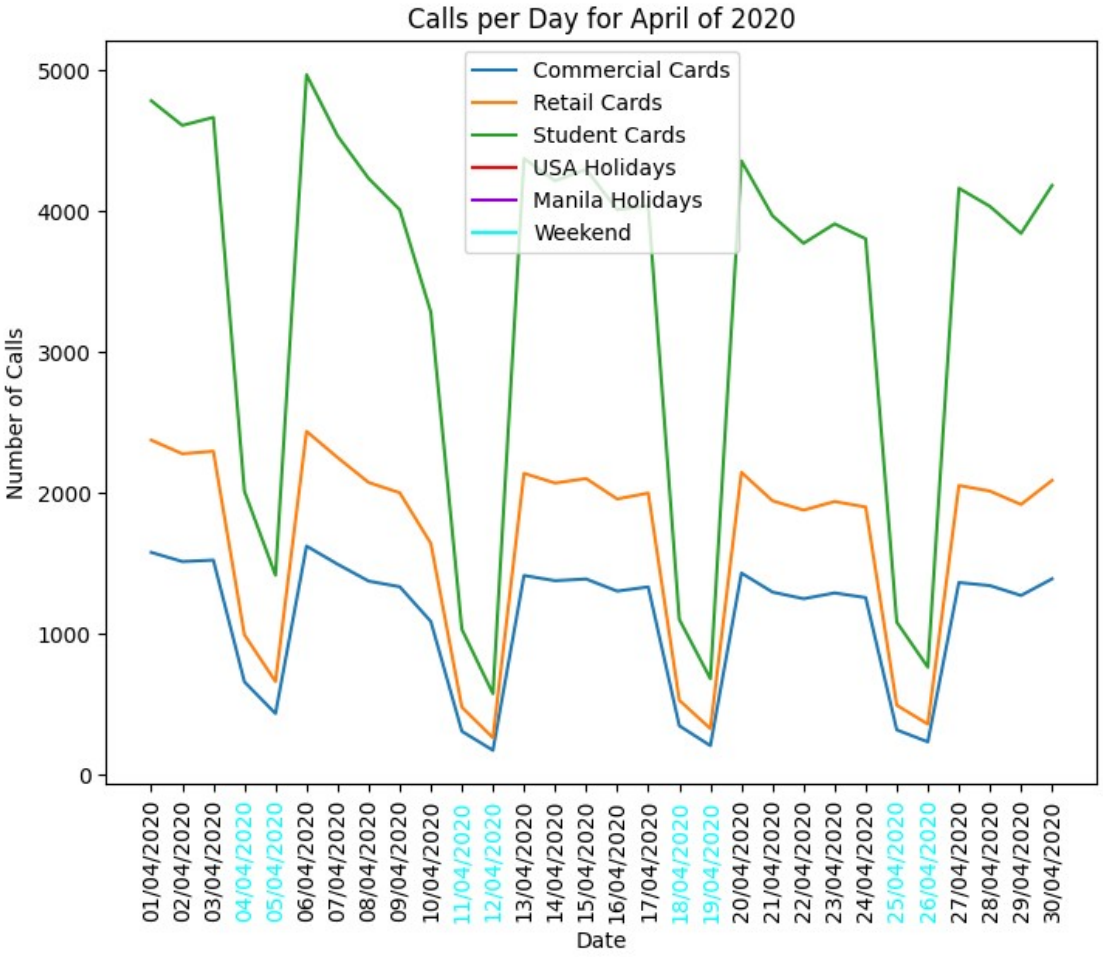

Calls per Day for November of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
7000
Veterans Day
Thanksgiving Day
Black Friday
Bonifacio
01/11/2019
02/11/2019
03/11/2019
04/11/2019
05/11/2019
06/11/2019
07/11/2019
08/11/2019
09/11/2019
10/11/2019
11/11/2019
12/11/2019
13/11/2019
14/11/2019
15/11/2019
16/11/2019
17/11/2019
18/11/2019
19/11/2019
20/11/2019
21/11/2019
22/11/2019
23/11/2019
24/11/2019
25/11/2019
26/11/2019
27/11/2019
28/11/2019
29/11/2019
30/11/2019
Date

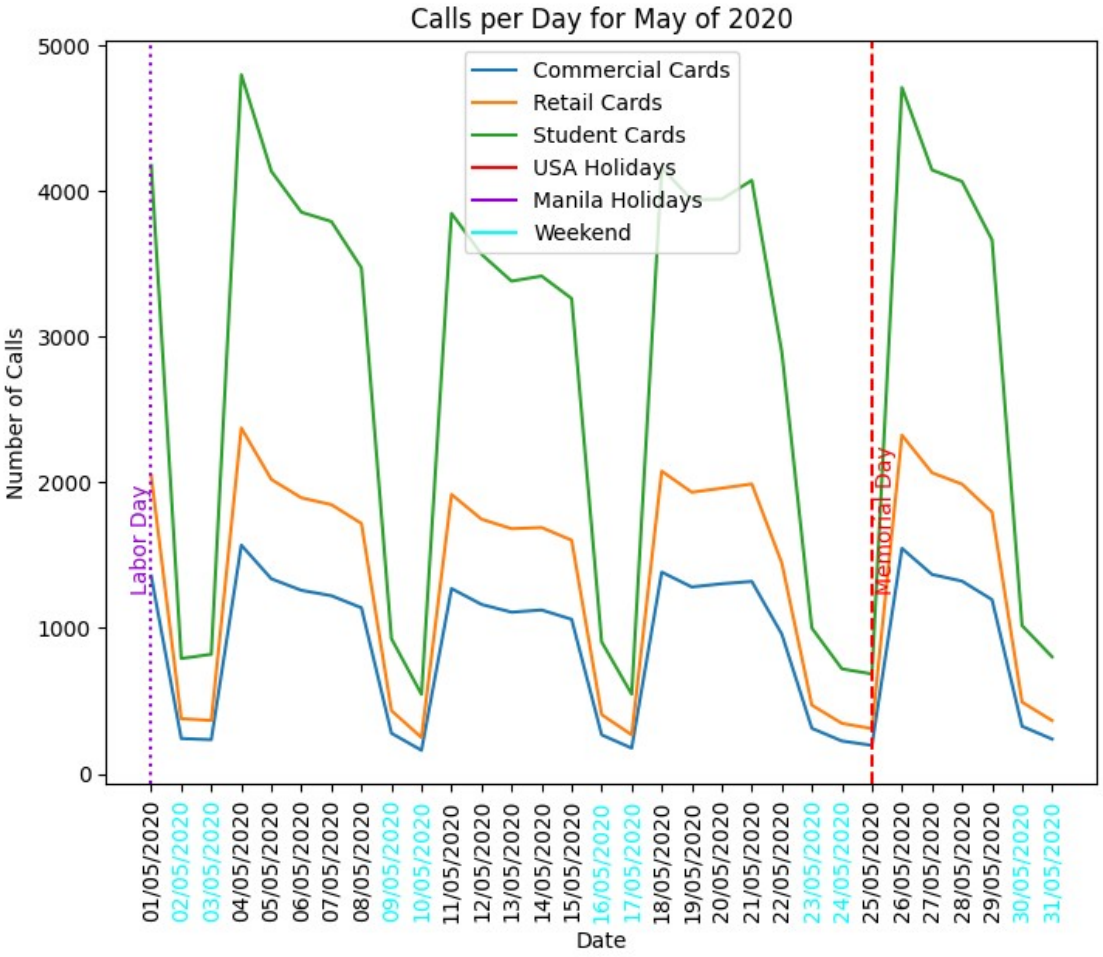

Calls per Day for December of 2019
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
Christmas Eve
Christmas Day
Rizal
01/12/2019
02/12/2019
03/12/2019
04/12/2019
05/12/2019
06/12/2019
07/12/2019
08/12/2019
09/12/2019
10/12/2019
11/12/2019
12/12/2019
13/12/2019
14/12/2019
15/12/2019
16/12/2019
17/12/2019
18/12/2019
19/12/2019
20/12/2019
21/12/2019
22/12/2019
23/12/2019
24/12/2019
25/12/2019
26/12/2019
27/12/2019
28/12/2019
29/12/2019
30/12/2019
31/12/2019
Date

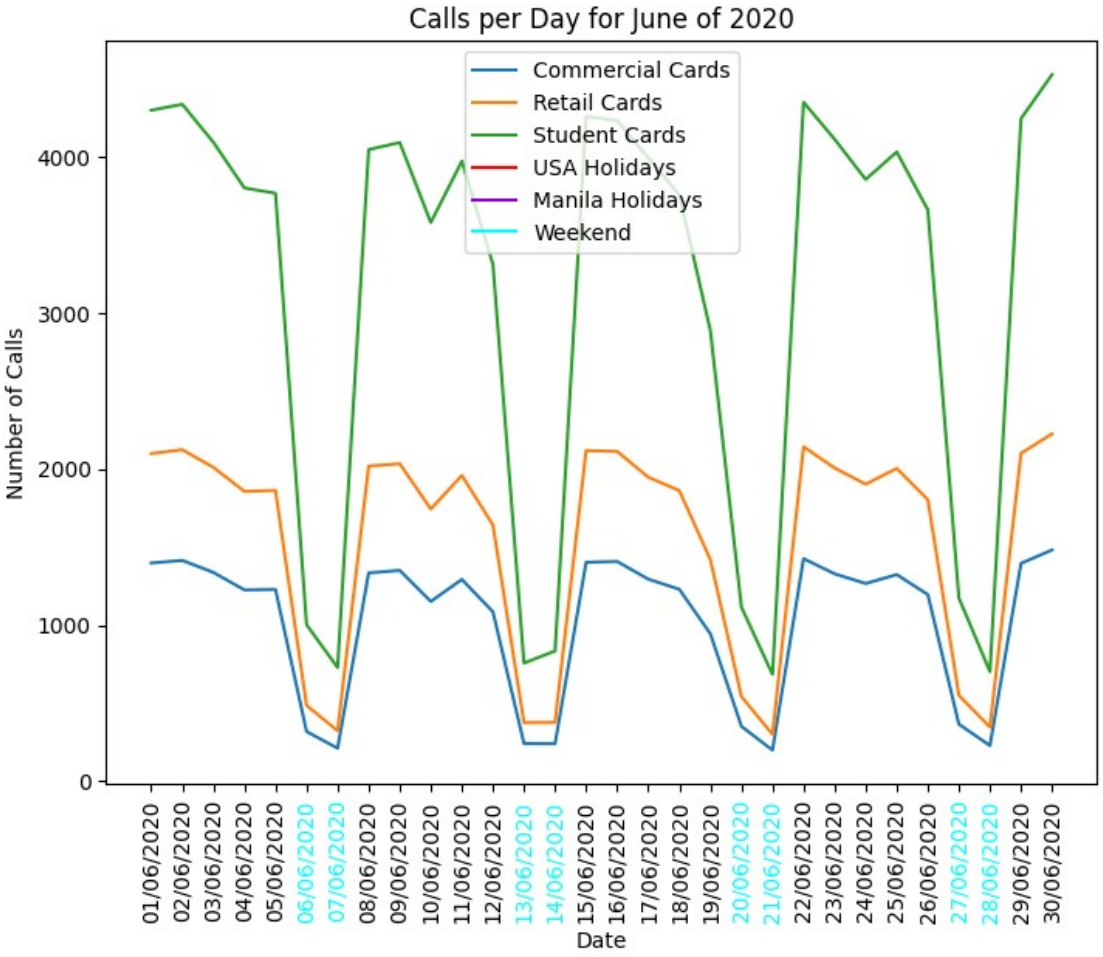

Calls per Day for January of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
Date
New Year
New Year's Day
Birthday of Martin Luther King, J.

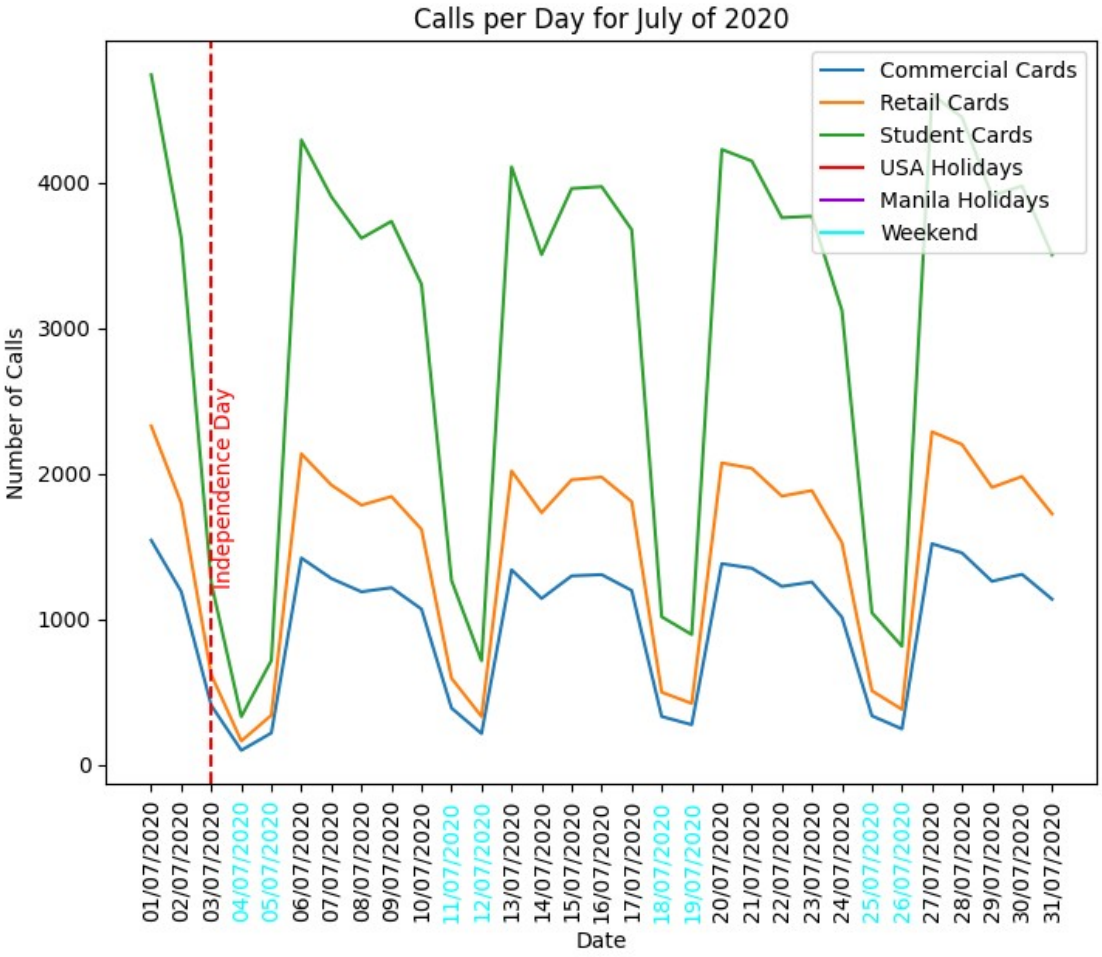

Calls per Day for February of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
1000
2000
3000
4000
5000
6000
Washington's Birthday
01/02/2020
02/02/2020
03/02/2020
04/02/2020
05/02/2020
06/02/2020
07/02/2020
08/02/2020
09/02/2020
10/02/2020
11/02/2020
12/02/2020
13/02/2020
14/02/2020
15/02/2020
16/02/2020
17/02/2020
18/02/2020
19/02/2020
20/02/2020
21/02/2020
22/02/2020
23/02/2020
24/02/2020
25/02/2020
26/02/2020
27/02/2020
28/02/2020
29/02/2020
Date

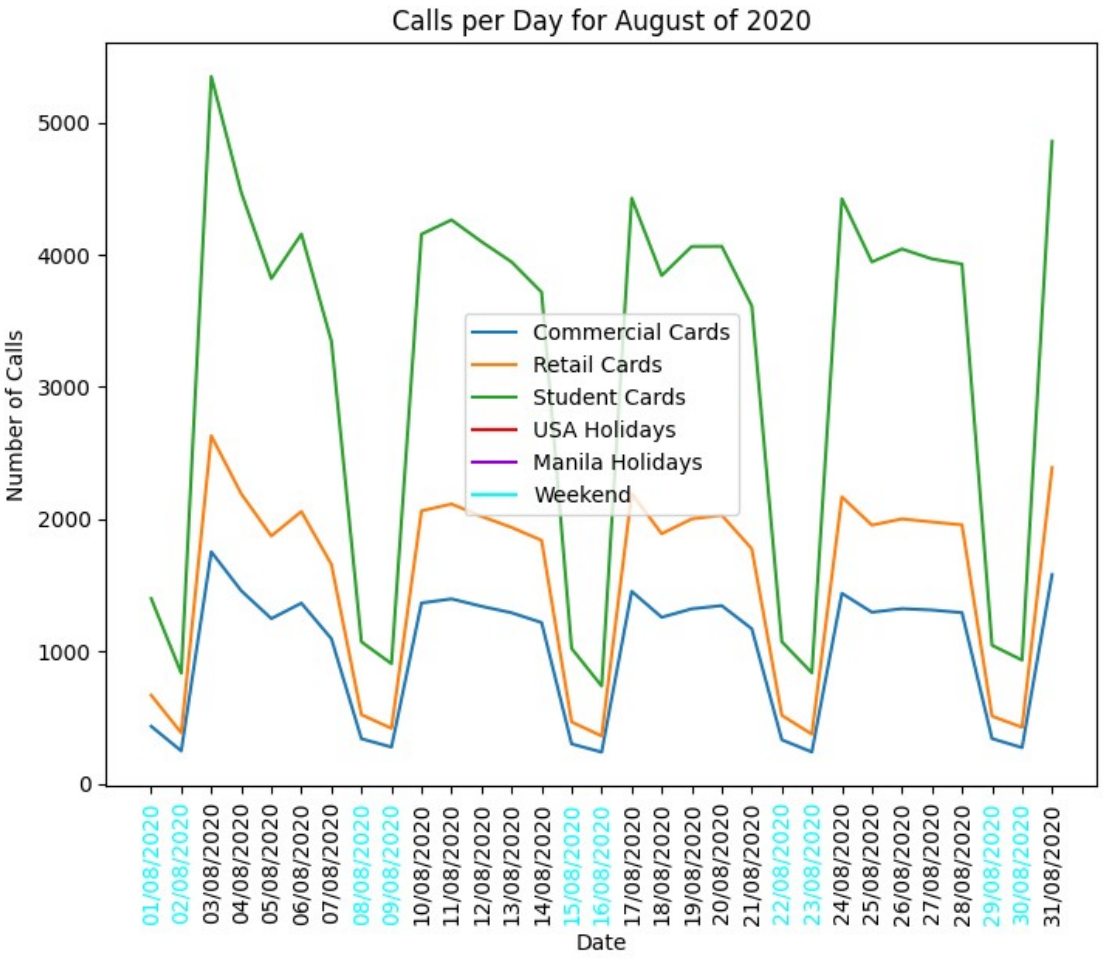

Calls per Day for March of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
6000
01/03/2020
02/03/2020
03/03/2020
04/03/2020
05/03/2020
06/03/2020
07/03/2020
08/03/2020
09/03/2020
10/03/2020
11/03/2020
12/03/2020
13/03/2020
14/03/2020
15/03/2020
16/03/2020
17/03/2020
18/03/2020
19/03/2020
20/03/2020
21/03/2020
22/03/2020
23/03/2020
24/03/2020
25/03/2020
26/03/2020
27/03/2020
28/03/2020
29/03/2020
30/03/2020
31/03/2020
Date

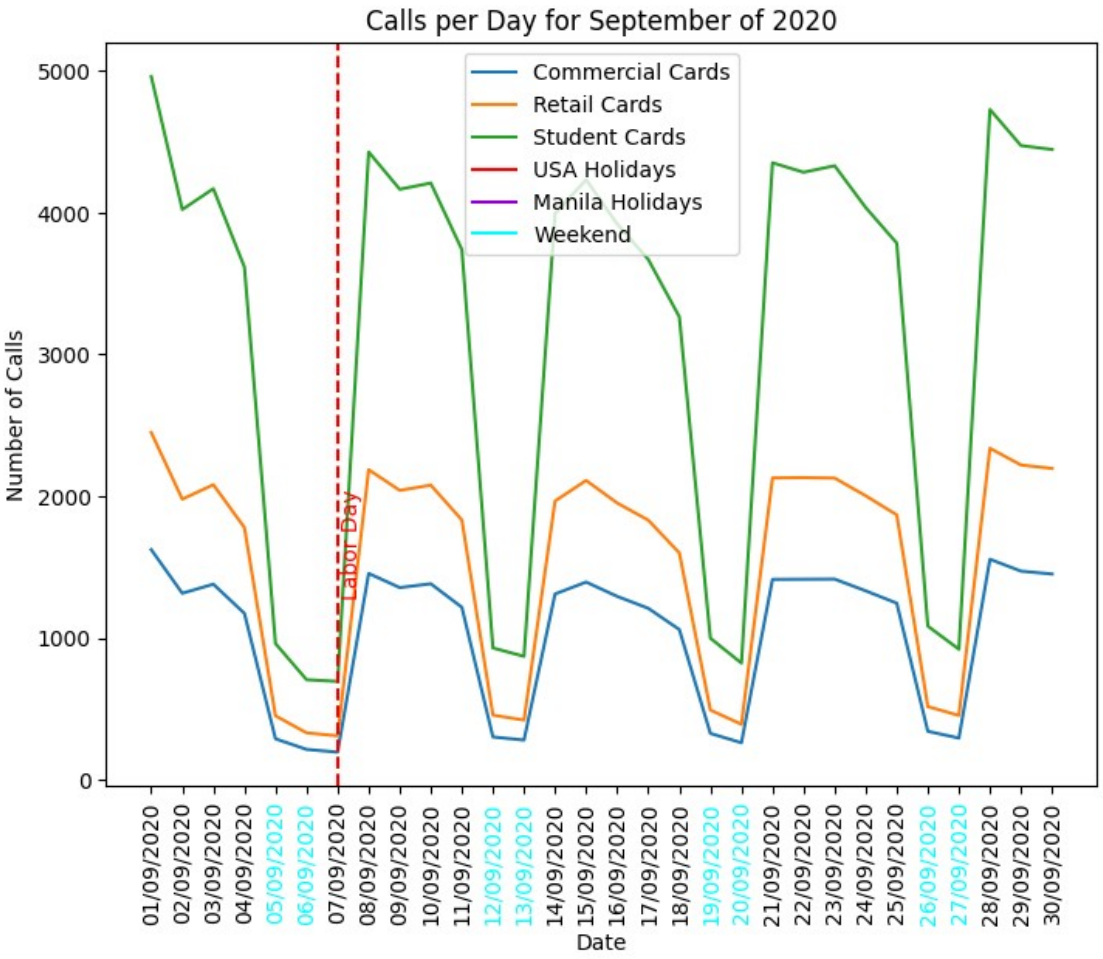

Calls per Day for April of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
01/04/2020
02/04/2020
03/04/2020
04/04/2020
05/04/2020
06/04/2020
07/04/2020
08/04/2020
09/04/2020
10/04/2020
11/04/2020
12/04/2020
13/04/2020
14/04/2020
15/04/2020
16/04/2020
17/04/2020
18/04/2020
19/04/2020
20/04/2020
21/04/2020
22/04/2020
23/04/2020
24/04/2020
25/04/2020
26/04/2020
27/04/2020
28/04/2020
29/04/2020
30/04/2020
Date

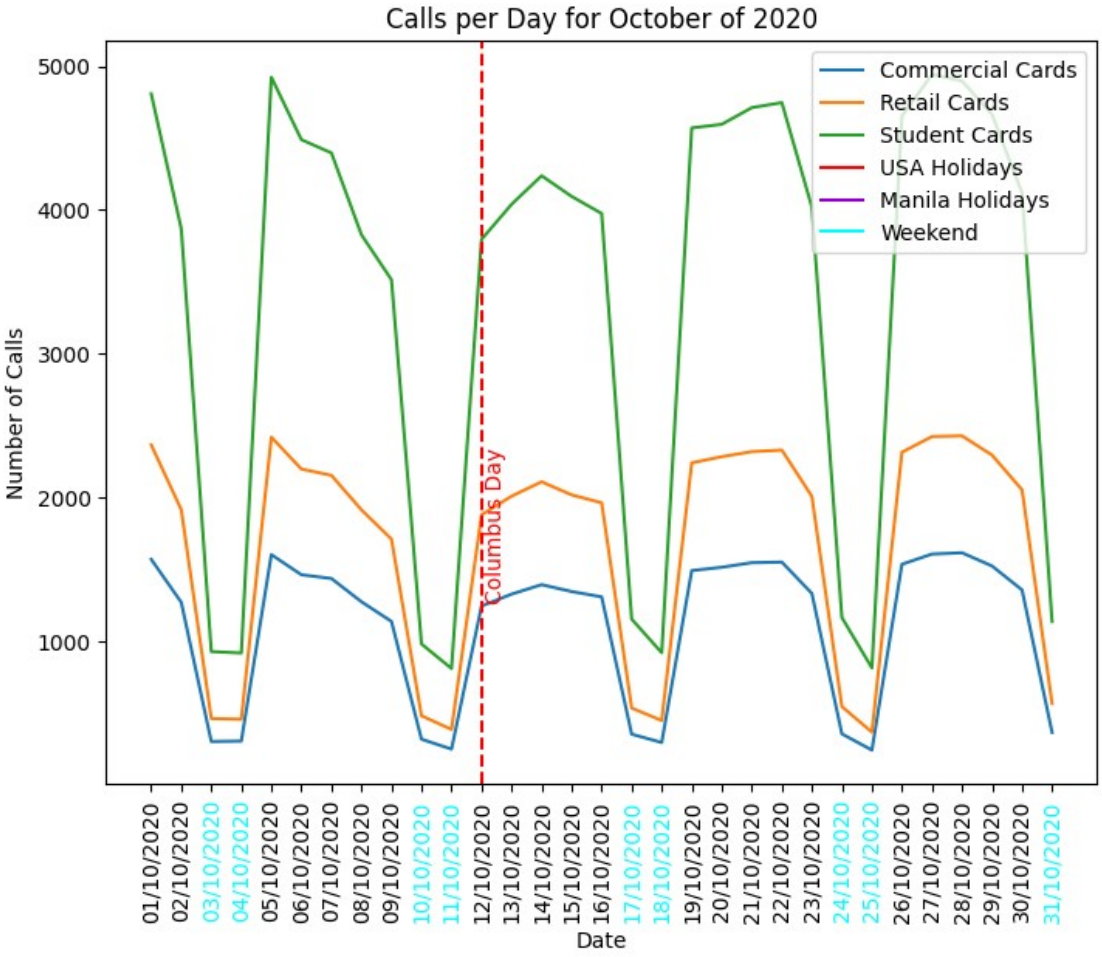

Calls per Day for May of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
5000
4000
3000
2000
1000
0
Labor Day
Memorial Day
01/05/2020
02/05/2020
03/05/2020
04/05/2020
05/05/2020
06/05/2020
07/05/2020
08/05/2020
09/05/2020
10/05/2020
11/05/2020
12/05/2020
13/05/2020
14/05/2020
15/05/2020
16/05/2020
17/05/2020
18/05/2020
19/05/2020
20/05/2020
21/05/2020
22/05/2020
23/05/2020
24/05/2020
25/05/2020
26/05/2020
27/05/2020
28/05/2020
29/05/2020
30/05/2020
31/05/2020
Date

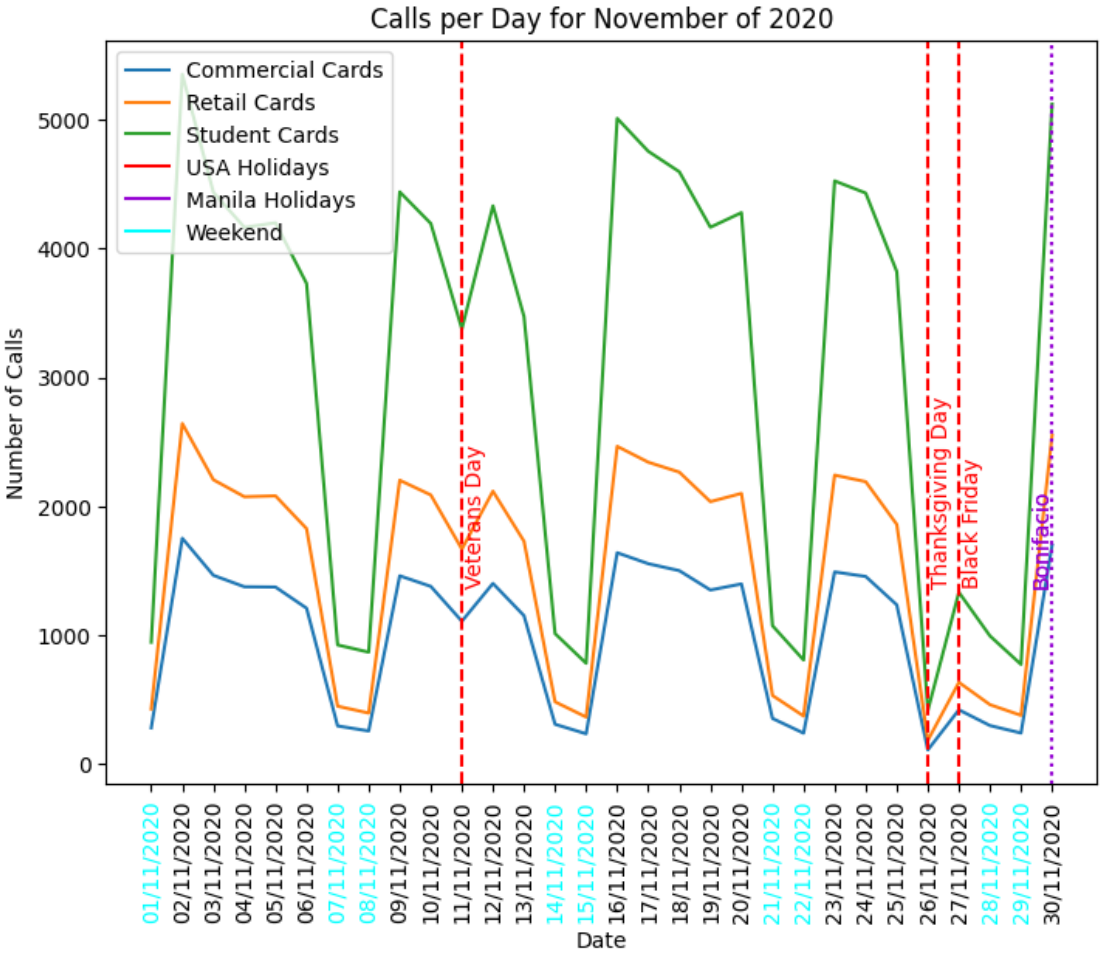

Calls per Day for June of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
01/06/2020
02/06/2020
03/06/2020
04/06/2020
05/06/2020
06/06/2020
07/06/2020
08/06/2020
09/06/2020
10/06/2020
11/06/2020
12/06/2020
13/06/2020
14/06/2020
15/06/2020
16/06/2020
17/06/2020
18/06/2020
19/06/2020
20/06/2020
21/06/2020
22/06/2020
23/06/2020
24/06/2020
25/06/2020
26/06/2020
27/06/2020
28/06/2020
29/06/2020
30/06/2020
Date

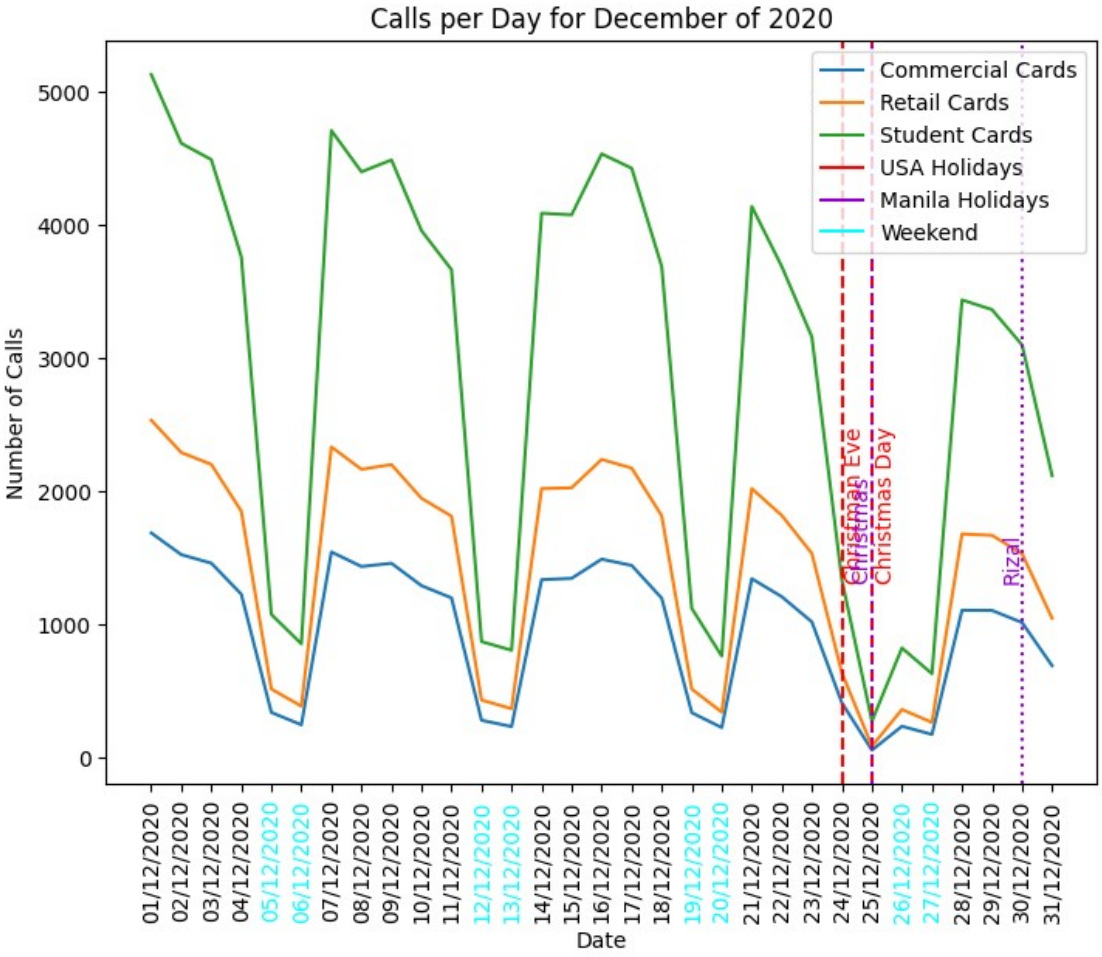

Calls per Day for July of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
Independence Day
01/07/2020
02/07/2020
03/07/2020
04/07/2020
05/07/2020
06/07/2020
07/07/2020
08/07/2020
09/07/2020
10/07/2020
11/07/2020
12/07/2020
13/07/2020
14/07/2020
15/07/2020
16/07/2020
17/07/2020
18/07/2020
19/07/2020
20/07/2020
21/07/2020
22/07/2020
23/07/2020
24/07/2020
25/07/2020
26/07/2020
27/07/2020
28/07/2020
29/07/2020
30/07/2020
31/07/2020
Date

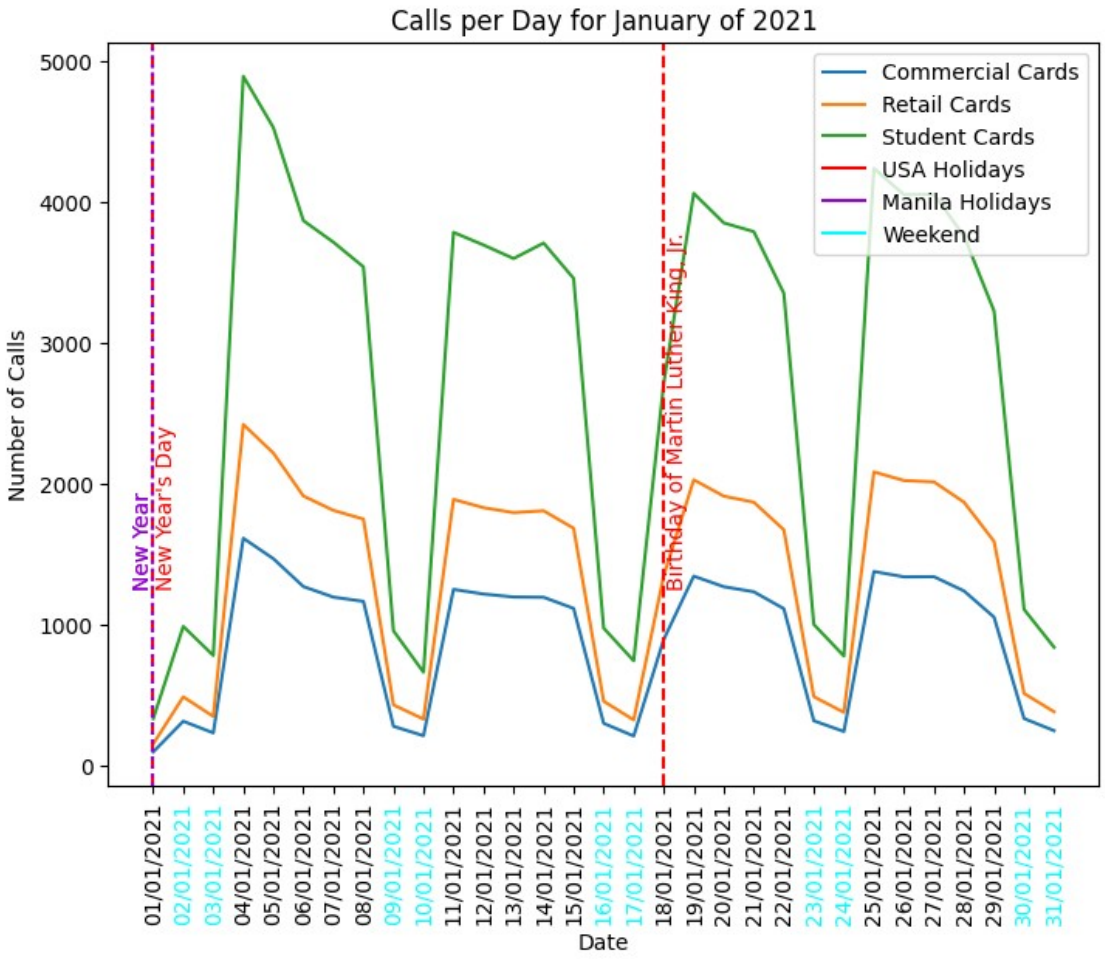

Calls per Day for August of 2020
Number of Calls
0
1000
2000
3000
4000
5000
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
01/08/2020
02/08/2020
03/08/2020
04/08/2020
05/08/2020
06/08/2020
07/08/2020
08/08/2020
09/08/2020
10/08/2020
11/08/2020
12/08/2020
13/08/2020
14/08/2020
15/08/2020
16/08/2020
17/08/2020
18/08/2020
19/08/2020
20/08/2020
21/08/2020
22/08/2020
23/08/2020
24/08/2020
25/08/2020
26/08/2020
27/08/2020
28/08/2020
29/08/2020
30/08/2020
31/08/2020
Date

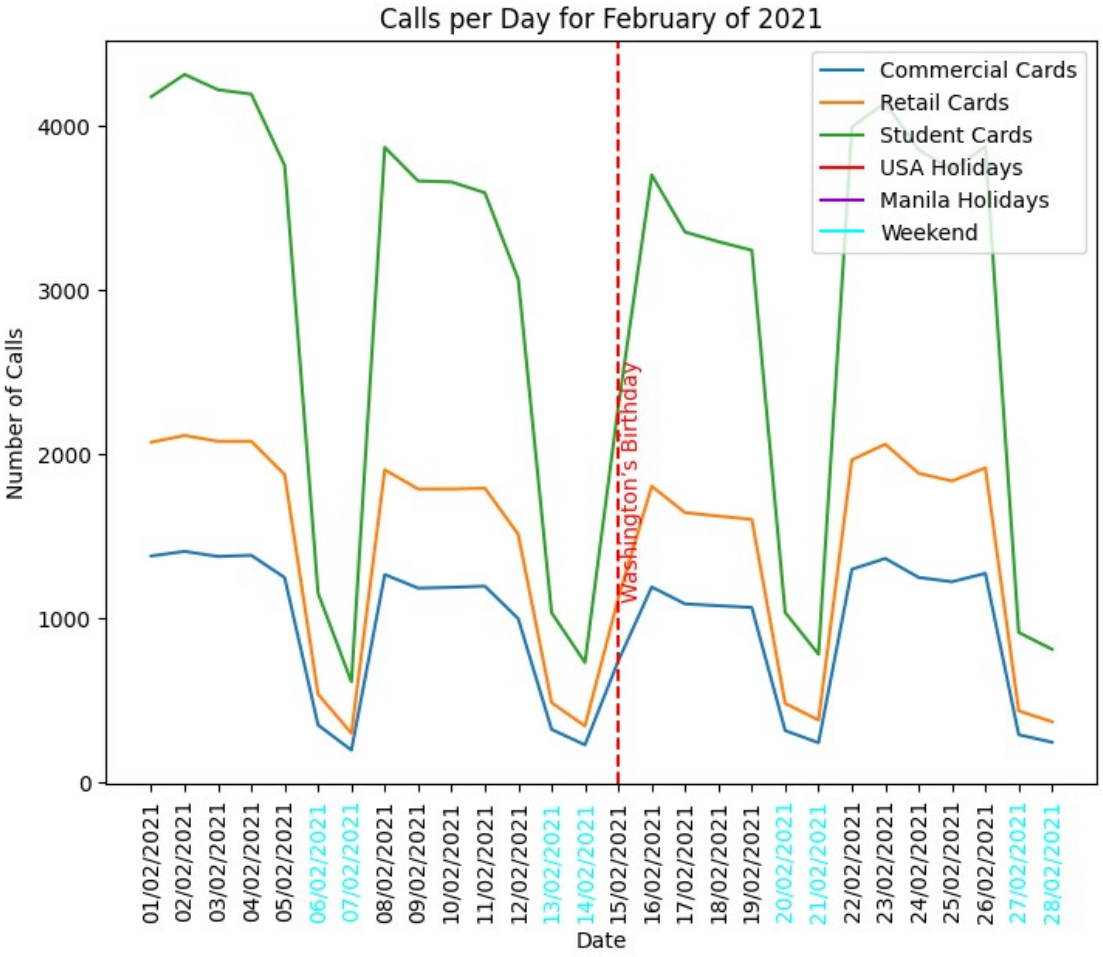

Calls per Day for September of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Labor Day
Number of Calls
5000
4000
3000
2000
1000
0
01/09/2020
02/09/2020
03/09/2020
04/09/2020
05/09/2020
06/09/2020
07/09/2020
08/09/2020
09/09/2020
10/09/2020
11/09/2020
12/09/2020
13/09/2020
14/09/2020
15/09/2020
16/09/2020
17/09/2020
18/09/2020
19/09/2020
20/09/2020
21/09/2020
22/09/2020
23/09/2020
24/09/2020
25/09/2020
26/09/2020
27/09/2020
28/09/2020
29/09/2020
30/09/2020
Date

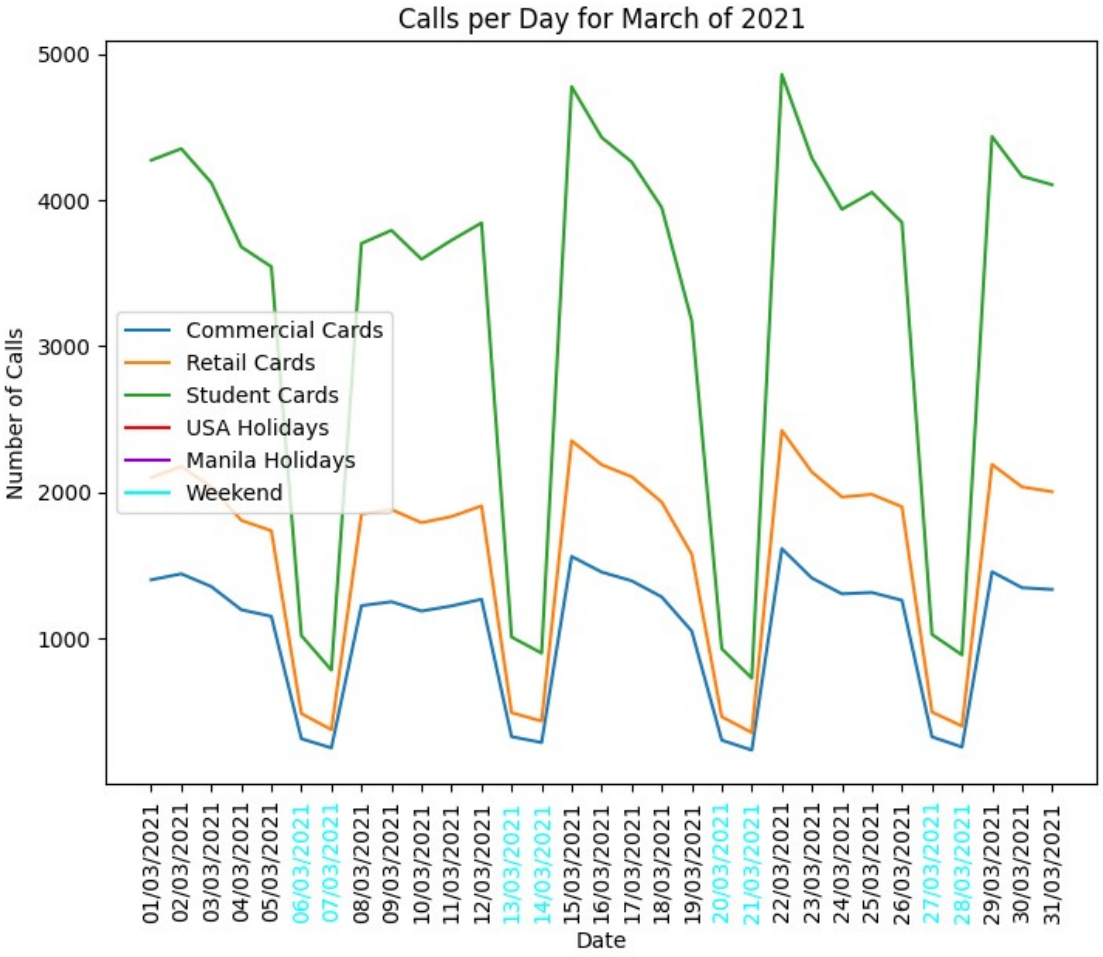

Calls per Day for October of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
5000
4000
3000
2000
1000
Columbus Day
01/10/2020
02/10/2020
03/10/2020
04/10/2020
05/10/2020
06/10/2020
07/10/2020
08/10/2020
09/10/2020
10/10/2020
11/10/2020
12/10/2020
13/10/2020
14/10/2020
15/10/2020
16/10/2020
17/10/2020
18/10/2020
19/10/2020
20/10/2020
21/10/2020
22/10/2020
23/10/2020
24/10/2020
25/10/2020
26/10/2020
27/10/2020
28/10/2020
29/10/2020
30/10/2020
31/10/2020
Date

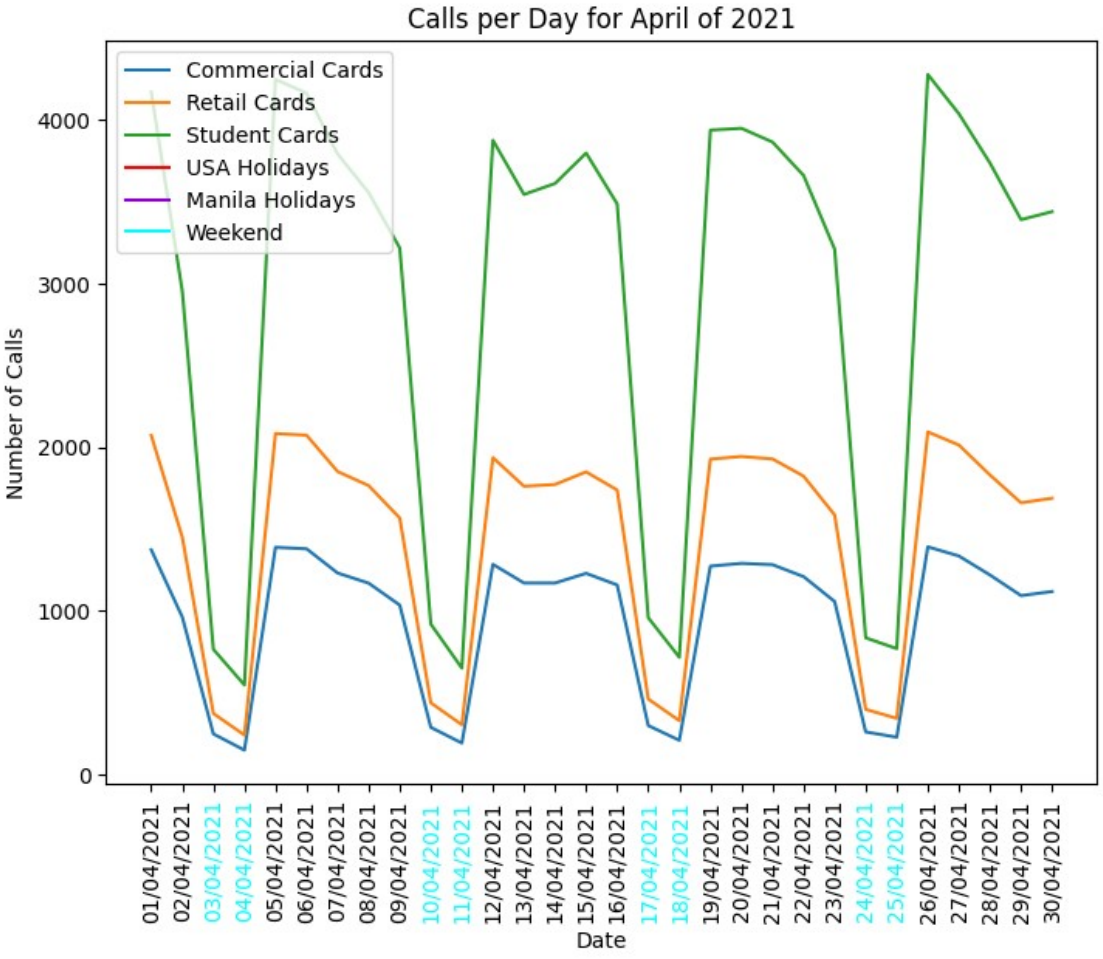

Calls per Day for November of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
Veterans Day
Thanksgiving Day
Black Friday
Bonifacio
01/11/2020
02/11/2020
03/11/2020
04/11/2020
05/11/2020
06/11/2020
07/11/2020
08/11/2020
09/11/2020
10/11/2020
11/11/2020
12/11/2020
13/11/2020
14/11/2020
15/11/2020
16/11/2020
17/11/2020
18/11/2020
19/11/2020
20/11/2020
21/11/2020
22/11/2020
23/11/2020
24/11/2020
25/11/2020
26/11/2020
27/11/2020
28/11/2020
29/11/2020
30/11/2020
Date


Calls per Day for December of 2020
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
Christmas Eve
Christmas
Christmas Day
Rizal
01/12/2020
02/12/2020
03/12/2020
04/12/2020
05/12/2020
06/12/2020
07/12/2020
08/12/2020
09/12/2020
10/12/2020
11/12/2020
12/12/2020
13/12/2020
14/12/2020
15/12/2020
16/12/2020
17/12/2020
18/12/2020
19/12/2020
20/12/2020
21/12/2020
22/12/2020
23/12/2020
24/12/2020
25/12/2020
26/12/2020
27/12/2020
28/12/2020
29/12/2020
30/12/2020
31/12/2020
Date


Calls per Day for January of 2021
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Number of Calls
0
1000
2000
3000
4000
5000
New Year
New Year's Day
Birthday of Martin Luther King, Jr.
01/01/2021
02/01/2021
03/01/2021
04/01/2021
05/01/2021
06/01/2021
07/01/2021
08/01/2021
09/01/2021
10/01/2021
11/01/2021
12/01/2021
13/01/2021
14/01/2021
15/01/2021
16/01/2021
17/01/2021
18/01/2021
19/01/2021
20/01/2021
21/01/2021
22/01/2021
23/01/2021
24/01/2021
25/01/2021
26/01/2021
27/01/2021
28/01/2021
29/01/2021
30/01/2021
31/01/2021
Date


Calls per Day for February of 2021
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
Washington's Birthday
Number of Calls
0
1000
2000
3000
4000
01/02/2021
02/02/2021
03/02/2021
04/02/2021
05/02/2021
06/02/2021
07/02/2021
08/02/2021
09/02/2021
10/02/2021
11/02/2021
12/02/2021
13/02/2021
14/02/2021
15/02/2021
16/02/2021
17/02/2021
18/02/2021
19/02/2021
20/02/2021
21/02/2021
22/02/2021
23/02/2021
24/02/2021
25/02/2021
26/02/2021
27/02/2021
28/02/2021
Date


Calls per Day for March of 2021
Number of Calls
5000
4000
3000
2000
1000
Commercial Cards
Retail Cards
Student Cards
USA Holidays
Manila Holidays
Weekend
01/03/2021
02/03/2021
03/03/2021
04/03/2021
05/03/2021
06/03/2021
07/03/2021
08/03/2021
09/03/2021
10/03/2021
11/03/2021
12/03/2021
13/03/2021
14/03/2021
15/03/2021
16/03/2021
17/03/2021
18/03/2021
19/03/2021
20/03/2021
21/03/2021
22/03/2021
23/03/2021
24/03/2021
25/03/2021
26/03/2021
27/03/2021
28/03/2021
29/03/2021
30/03/2021
31/03/2021
Date



Prediction

```
dates17 = df_17.date.values
dates_not17 = df_not17_sorted.date.values
all_dates = list(dates17)
all_dates.extend(list(dates_not17))
# print(all_dates)
all_dates = pd.to_datetime(all_dates, format="%d/%m/%Y")
# print(all_dates)

cc17 = df_17.cc.values
cc_not17 = df_not17_sorted.cc.values
all_cc = list(cc17)
all_cc.extend(cc_not17)
# print(all_cc)

lstm_data = pd.DataFrame({
    "ds": all_dates,
    "y": all_cc
})
lstm_data
```

```
             ds       y
0    2017-01-01   264.0
1    2017-01-02   528.0
2    2017-01-03  2505.0
3    2017-01-04  2145.0
4    2017-01-05  2164.0
...         ...     ...
1576 2021-04-26  1392.0
1577 2021-04-27  1336.0
1578 2021-04-28  1220.0
1579 2021-04-29  1094.0
1580 2021-04-30  1119.0

[1581 rows x 2 columns]
```

```
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten

# preparing independent and dependent features
def prepare_data(timeseries_data, n_features):
     X, y =[],[]
     for i in range(len(timeseries_data)):
          # find the end of this pattern
          end_ix = i + n_features
          # check if we are beyond the sequence
          if end_ix > len(timeseries_data)-1:
               break
          # gather input and output parts of the pattern
          seq_x, seq_y = timeseries_data[i:end_ix],
timeseries_data[end_ix]
          X.append(seq_x)
          y.append(seq_y)
     return np.array(X), np.array(y)

# define input sequence
timeseries_data = all_cc
# choose a number of time steps
n_steps = 7
# split into samples
X, y = prepare_data(timeseries_data, n_steps)

# reshape from [samples, timesteps] into [samples, timesteps,
features]
n_features = 1
X = X.reshape((X.shape[0], X.shape[1], n_features))

# define model
model = Sequential()
```

```
model.add(LSTM(50, activation='relu', return_sequences=True,
input_shape=(n_steps, n_features)))
model.add(LSTM(50, activation='relu'))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
# fit model
model.fit(X, y, epochs=500, verbose=1)
```

```
Epoch 1/500
50/50 [==============================] - 2s 3ms/step - loss:
1113896.0000
Epoch 2/500
50/50 [==============================] - 0s 3ms/step - loss:
195022.7969
Epoch 3/500
50/50 [==============================] - 0s 3ms/step - loss:
136563.2969
Epoch 4/500
50/50 [==============================] - 0s 3ms/step - loss:
118197.0859
Epoch 5/500
50/50 [==============================] - 0s 3ms/step - loss:
104599.4922
Epoch 6/500
50/50 [==============================] - 0s 3ms/step - loss:
177772.3594
Epoch 7/500
50/50 [==============================] - 0s 3ms/step - loss:
181992.1094
Epoch 8/500
50/50 [==============================] - 0s 3ms/step - loss:
208646.7500
Epoch 9/500
50/50 [==============================] - 0s 3ms/step - loss:
236557.4375
Epoch 10/500
50/50 [==============================] - 0s 3ms/step - loss:
154889.8125
Epoch 11/500
50/50 [==============================] - 0s 3ms/step - loss:
138113.7344
Epoch 12/500
50/50 [==============================] - 0s 3ms/step - loss:
139994.6562
Epoch 13/500
50/50 [==============================] - 0s 3ms/step - loss:
136935.1719
Epoch 14/500
50/50 [==============================] - 0s 3ms/step - loss:
126615.7266
```

```
Epoch 15/500
50/50 [==============================] - 0s 3ms/step - loss:
109311.1328
Epoch 16/500
50/50 [==============================] - 0s 3ms/step - loss:
101247.6250
Epoch 17/500
50/50 [==============================] - 0s 3ms/step - loss:
98932.4297
Epoch 18/500
50/50 [==============================] - 0s 3ms/step - loss:
97167.6328
Epoch 19/500
50/50 [==============================] - 0s 3ms/step - loss:
103274.7656
Epoch 20/500
50/50 [==============================] - 0s 3ms/step - loss:
98956.2266
Epoch 21/500
50/50 [==============================] - 0s 3ms/step - loss:
103466.6719
Epoch 22/500
50/50 [==============================] - 0s 3ms/step - loss:
100932.4766
Epoch 23/500
50/50 [==============================] - 0s 3ms/step - loss:
94971.2734
Epoch 24/500
50/50 [==============================] - 0s 3ms/step - loss:
98537.5312
Epoch 25/500
50/50 [==============================] - 0s 3ms/step - loss:
93551.8359
Epoch 26/500
50/50 [==============================] - 0s 3ms/step - loss:
99746.5781
Epoch 27/500
50/50 [==============================] - 0s 3ms/step - loss:
108707.4375
Epoch 28/500
50/50 [==============================] - 0s 3ms/step - loss:
95745.2422
Epoch 29/500
50/50 [==============================] - 0s 3ms/step - loss:
93859.4141
Epoch 30/500
50/50 [==============================] - 0s 3ms/step - loss:
87338.0625
Epoch 31/500
50/50 [==============================] - 0s 3ms/step - loss:
```

```
89907.8359
Epoch 32/500
50/50 [==============================] - 0s 3ms/step - loss:
93018.9141
Epoch 33/500
50/50 [==============================] - 0s 3ms/step - loss:
85993.1016
Epoch 34/500
50/50 [==============================] - 0s 3ms/step - loss:
90065.1719
Epoch 35/500
50/50 [==============================] - 0s 3ms/step - loss:
88101.5469
Epoch 36/500
50/50 [==============================] - 0s 3ms/step - loss:
89085.3906
Epoch 37/500
50/50 [==============================] - 0s 3ms/step - loss:
88349.7344
Epoch 38/500
50/50 [==============================] - 0s 3ms/step - loss:
84216.1094
Epoch 39/500
50/50 [==============================] - 0s 3ms/step - loss:
90629.8359
Epoch 40/500
50/50 [==============================] - 0s 3ms/step - loss:
93599.0156
Epoch 41/500
50/50 [==============================] - 0s 3ms/step - loss:
88702.8984
Epoch 42/500
50/50 [==============================] - 0s 3ms/step - loss:
93061.6406
Epoch 43/500
50/50 [==============================] - 0s 3ms/step - loss:
83411.6172
Epoch 44/500
50/50 [==============================] - 0s 3ms/step - loss:
83725.5156
Epoch 45/500
50/50 [==============================] - 0s 3ms/step - loss:
87212.5781
Epoch 46/500
50/50 [==============================] - 0s 3ms/step - loss:
92420.4531
Epoch 47/500
50/50 [==============================] - 0s 3ms/step - loss:
92949.0859
Epoch 48/500
```

```
50/50 [==============================] - 0s 3ms/step - loss:
90891.3203
Epoch 49/500
50/50 [==============================] - 0s 3ms/step - loss:
92886.4453
Epoch 50/500
50/50 [==============================] - 0s 3ms/step - loss:
96329.6172
Epoch 51/500
50/50 [==============================] - 0s 3ms/step - loss:
88877.4922
Epoch 52/500
50/50 [==============================] - 0s 3ms/step - loss:
89462.2188
Epoch 53/500
50/50 [==============================] - 0s 3ms/step - loss:
91325.3125
Epoch 54/500
50/50 [==============================] - 0s 3ms/step - loss:
91177.2031
Epoch 55/500
50/50 [==============================] - 0s 3ms/step - loss:
87200.4062
Epoch 56/500
50/50 [==============================] - 0s 3ms/step - loss:
91140.1250
Epoch 57/500
50/50 [==============================] - 0s 3ms/step - loss:
87608.9531
Epoch 58/500
50/50 [==============================] - 0s 3ms/step - loss:
88015.9297
Epoch 59/500
50/50 [==============================] - 0s 3ms/step - loss:
106718.4688
Epoch 60/500
50/50 [==============================] - 0s 3ms/step - loss:
89783.9062
Epoch 61/500
50/50 [==============================] - 0s 3ms/step - loss:
115887.0625
Epoch 62/500
50/50 [==============================] - 0s 3ms/step - loss:
123937.5234
Epoch 63/500
50/50 [==============================] - 0s 3ms/step - loss:
122897.4141
Epoch 64/500
50/50 [==============================] - 0s 3ms/step - loss:
137507.4688
```

```
Epoch 65/500
50/50 [==============================] - 0s 3ms/step - loss:
173537.3281
Epoch 66/500
50/50 [==============================] - 0s 3ms/step - loss:
133203.3281
Epoch 67/500
50/50 [==============================] - 0s 3ms/step - loss:
114230.3906
Epoch 68/500
50/50 [==============================] - 0s 3ms/step - loss:
110540.3594
Epoch 69/500
50/50 [==============================] - 0s 3ms/step - loss:
108054.7266
Epoch 70/500
50/50 [==============================] - 0s 3ms/step - loss:
103882.4688
Epoch 71/500
50/50 [==============================] - 0s 3ms/step - loss:
101692.2109
Epoch 72/500
50/50 [==============================] - 0s 3ms/step - loss:
103533.8828
Epoch 73/500
50/50 [==============================] - 0s 3ms/step - loss:
100125.5312
Epoch 74/500
50/50 [==============================] - 0s 3ms/step - loss:
91417.9219
Epoch 75/500
50/50 [==============================] - 0s 3ms/step - loss:
91233.2266
Epoch 76/500
50/50 [==============================] - 0s 3ms/step - loss:
89865.5312
Epoch 77/500
50/50 [==============================] - 0s 3ms/step - loss:
97578.3047
Epoch 78/500
50/50 [==============================] - 0s 3ms/step - loss:
99155.8516
Epoch 79/500
50/50 [==============================] - 0s 3ms/step - loss:
91833.9609
Epoch 80/500
50/50 [==============================] - 0s 3ms/step - loss:
92436.1953
Epoch 81/500
50/50 [==============================] - 0s 3ms/step - loss:
```

```
91920.1797
Epoch 82/500
50/50 [==============================] - 0s 3ms/step - loss:
91850.6641
Epoch 83/500
50/50 [==============================] - 0s 3ms/step - loss:
90216.8672
Epoch 84/500
50/50 [==============================] - 0s 3ms/step - loss:
92173.6406
Epoch 85/500
50/50 [==============================] - 0s 3ms/step - loss:
105668.6328
Epoch 86/500
50/50 [==============================] - 0s 3ms/step - loss:
95578.6562
Epoch 87/500
50/50 [==============================] - 0s 3ms/step - loss:
107282.6719
Epoch 88/500
50/50 [==============================] - 0s 3ms/step - loss:
100617.8438
Epoch 89/500
50/50 [==============================] - 0s 3ms/step - loss:
96417.7188
Epoch 90/500
50/50 [==============================] - 0s 3ms/step - loss:
111101.8750
Epoch 91/500
50/50 [==============================] - 0s 3ms/step - loss:
94822.4766
Epoch 92/500
50/50 [==============================] - 0s 3ms/step - loss:
94036.3125
Epoch 93/500
50/50 [==============================] - 0s 3ms/step - loss:
92529.4531
Epoch 94/500
50/50 [==============================] - 0s 3ms/step - loss:
86105.1094
Epoch 95/500
50/50 [==============================] - 0s 3ms/step - loss:
90966.9531
Epoch 96/500
50/50 [==============================] - 0s 3ms/step - loss:
87175.5391
Epoch 97/500

50/50 [==============================] - 0s 3ms/step - loss:
87790.7734
```

```
Epoch 98/500
50/50 [==============================] - 0s 3ms/step - loss:
95836.8359
Epoch 99/500
50/50 [==============================] - 0s 3ms/step - loss:
92070.5703
Epoch 100/500
50/50 [==============================] - 0s 3ms/step - loss:
88539.7578
Epoch 101/500
50/50 [==============================] - 0s 3ms/step - loss:
86123.0391
Epoch 102/500
50/50 [==============================] - 0s 3ms/step - loss:
86980.2422
Epoch 103/500
50/50 [==============================] - 0s 3ms/step - loss:
86380.9062
Epoch 104/500
50/50 [==============================] - 0s 3ms/step - loss:
85733.5625
Epoch 105/500
50/50 [==============================] - 0s 3ms/step - loss:
82195.7344
Epoch 106/500
50/50 [==============================] - 0s 3ms/step - loss:
82278.3047
Epoch 107/500
50/50 [==============================] - 0s 3ms/step - loss:
81558.0469
Epoch 108/500
50/50 [==============================] - 0s 3ms/step - loss:
83897.0312
Epoch 109/500
50/50 [==============================] - 0s 3ms/step - loss:
82551.6719
Epoch 110/500
50/50 [==============================] - 0s 3ms/step - loss:
81033.3125
Epoch 111/500
50/50 [==============================] - 0s 3ms/step - loss:
82052.5469
Epoch 112/500
50/50 [==============================] - 0s 3ms/step - loss:
80582.6875
Epoch 113/500
50/50 [==============================] - 0s 3ms/step - loss:
80526.2500
Epoch 114/500
50/50 [==============================] - 0s 3ms/step - loss:
```

```
151400.3594
Epoch 115/500
50/50 [==============================] - 0s 3ms/step - loss:
111225.7500
Epoch 116/500
50/50 [==============================] - 0s 3ms/step - loss:
107276.4453
Epoch 117/500
50/50 [==============================] - 0s 3ms/step - loss:
99188.5547
Epoch 118/500
50/50 [==============================] - 0s 3ms/step - loss:
92717.0391
Epoch 119/500
50/50 [==============================] - 0s 3ms/step - loss:
92085.5703
Epoch 120/500
50/50 [==============================] - 0s 3ms/step - loss:
90990.8594
Epoch 121/500
50/50 [==============================] - 0s 3ms/step - loss:
89749.1094
Epoch 122/500
50/50 [==============================] - 0s 3ms/step - loss:
87269.7344
Epoch 123/500
50/50 [==============================] - 0s 3ms/step - loss:
86295.3359
Epoch 124/500
50/50 [==============================] - 0s 3ms/step - loss:
83974.9141
Epoch 125/500
50/50 [==============================] - 0s 3ms/step - loss:
81234.4688
Epoch 126/500
50/50 [==============================] - 0s 3ms/step - loss:
81005.0391
Epoch 127/500
50/50 [==============================] - 0s 3ms/step - loss:
78591.2578
Epoch 128/500
50/50 [==============================] - 0s 3ms/step - loss:
75836.2266
Epoch 129/500
50/50 [==============================] - 0s 3ms/step - loss:
74756.7812
Epoch 130/500
50/50 [==============================] - 0s 3ms/step - loss:
77278.5078
Epoch 131/500
```

```
50/50 [==============================] - 0s 3ms/step - loss:
73977.4297
Epoch 132/500
50/50 [==============================] - 0s 3ms/step - loss:
75407.0938
Epoch 133/500
50/50 [==============================] - 0s 3ms/step - loss:
77371.6562
Epoch 134/500
50/50 [==============================] - 0s 3ms/step - loss:
75932.5703
Epoch 135/500
50/50 [==============================] - 0s 3ms/step - loss:
80246.3281
Epoch 136/500
50/50 [==============================] - 0s 3ms/step - loss:
81042.1016
Epoch 137/500
50/50 [==============================] - 0s 3ms/step - loss:
73258.6641
Epoch 138/500
50/50 [==============================] - 0s 3ms/step - loss:
72889.4453
Epoch 139/500
50/50 [==============================] - 0s 3ms/step - loss:
72747.9453
Epoch 140/500
50/50 [==============================] - 0s 3ms/step - loss:
72602.0859
Epoch 141/500
50/50 [==============================] - 0s 3ms/step - loss:
71413.0156
Epoch 142/500
50/50 [==============================] - 0s 3ms/step - loss:
68528.8438
Epoch 143/500
50/50 [==============================] - 0s 3ms/step - loss:
71513.5938
Epoch 144/500
50/50 [==============================] - 0s 3ms/step - loss:
76260.9219
Epoch 145/500
50/50 [==============================] - 0s 3ms/step - loss:
78370.6562
Epoch 146/500
50/50 [==============================] - 0s 3ms/step - loss:
68628.2266
Epoch 147/500
50/50 [==============================] - 0s 3ms/step - loss:
70423.9453
```

```
Epoch 148/500
50/50 [==============================] - 0s 3ms/step - loss:
69218.4375
Epoch 149/500
50/50 [==============================] - 0s 3ms/step - loss:
69192.3125
Epoch 150/500
50/50 [==============================] - 0s 3ms/step - loss:
69013.6953
Epoch 151/500
50/50 [==============================] - 0s 3ms/step - loss:
70837.2031
Epoch 152/500
50/50 [==============================] - 0s 3ms/step - loss:
73026.7969
Epoch 153/500
50/50 [==============================] - 0s 3ms/step - loss:
67946.5078
Epoch 154/500
50/50 [==============================] - 0s 3ms/step - loss:
65295.7109
Epoch 155/500
50/50 [==============================] - 0s 3ms/step - loss:
68216.9375
Epoch 156/500
50/50 [==============================] - 0s 3ms/step - loss:
66146.4766
Epoch 157/500
50/50 [==============================] - 0s 3ms/step - loss:
67786.1719
Epoch 158/500
50/50 [==============================] - 0s 3ms/step - loss:
66701.1016
Epoch 159/500
50/50 [==============================] - 0s 3ms/step - loss:
64647.1562
Epoch 160/500
50/50 [==============================] - 0s 3ms/step - loss:
70723.2969
Epoch 161/500
50/50 [==============================] - 0s 3ms/step - loss:
70822.5859
Epoch 162/500
50/50 [==============================] - 0s 3ms/step - loss:
64331.9648
Epoch 163/500
50/50 [==============================] - 0s 3ms/step - loss:
65686.6016
Epoch 164/500
50/50 [==============================] - 0s 3ms/step - loss:
```

```
62415.6797
Epoch 165/500
50/50 [==============================] - 0s 3ms/step - loss:
62610.6367
Epoch 166/500
50/50 [==============================] - 0s 3ms/step - loss:
66478.7969
Epoch 167/500
50/50 [==============================] - 0s 3ms/step - loss:
66925.0078
Epoch 168/500
50/50 [==============================] - 0s 3ms/step - loss:
62333.9062
Epoch 169/500
50/50 [==============================] - 0s 3ms/step - loss:
64400.8320
Epoch 170/500
50/50 [==============================] - 0s 3ms/step - loss:
82322.4688
Epoch 171/500
50/50 [==============================] - 0s 3ms/step - loss:
79206.2500
Epoch 172/500
50/50 [==============================] - 0s 3ms/step - loss:
76665.9375
Epoch 173/500
50/50 [==============================] - 0s 3ms/step - loss:
75738.1250
Epoch 174/500
50/50 [==============================] - 0s 3ms/step - loss:
71826.9141
Epoch 175/500
50/50 [==============================] - 0s 3ms/step - loss:
67113.5938
Epoch 176/500
50/50 [==============================] - 0s 3ms/step - loss:
67577.8203
Epoch 177/500
50/50 [==============================] - 0s 3ms/step - loss:
64545.7344
Epoch 178/500
50/50 [==============================] - 0s 3ms/step - loss:
62253.7695
Epoch 179/500
50/50 [==============================] - 0s 3ms/step - loss:
64256.6719
Epoch 180/500
50/50 [==============================] - 0s 3ms/step - loss:
75179.8203
Epoch 181/500
```

```
50/50 [==============================] - 0s 3ms/step - loss:
70668.6875
Epoch 182/500
50/50 [==============================] - 0s 3ms/step - loss:
75449.6641
Epoch 183/500
50/50 [==============================] - 0s 3ms/step - loss:
73071.2188
Epoch 184/500
50/50 [==============================] - 0s 3ms/step - loss:
69285.7969
Epoch 185/500
50/50 [==============================] - 0s 3ms/step - loss:
77096.5547
Epoch 186/500
50/50 [==============================] - 0s 3ms/step - loss:
65299.8359
Epoch 187/500
50/50 [==============================] - 0s 3ms/step - loss:
63036.9219
Epoch 188/500
50/50 [==============================] - 0s 3ms/step - loss:
63543.9531
Epoch 189/500
50/50 [==============================] - 0s 3ms/step - loss:
66185.5312
Epoch 190/500
50/50 [==============================] - 0s 3ms/step - loss:
62241.3320
Epoch 191/500
50/50 [==============================] - 0s 3ms/step - loss:
60393.1992
Epoch 192/500

50/50 [==============================] - 0s 3ms/step - loss:
62531.1094
Epoch 193/500
50/50 [==============================] - 0s 3ms/step - loss:
62554.2891
Epoch 194/500
50/50 [==============================] - 0s 3ms/step - loss:
60903.7461
Epoch 195/500
50/50 [==============================] - 0s 3ms/step - loss:
62881.3516
Epoch 196/500
50/50 [==============================] - 0s 3ms/step - loss:
60029.3047
Epoch 197/500
50/50 [==============================] - 0s 3ms/step - loss:
```

```
64028.3945
Epoch 198/500
50/50 [==============================] - 0s 3ms/step - loss:
90664.1797
Epoch 199/500
50/50 [==============================] - 0s 3ms/step - loss:
87862.4375
Epoch 200/500
50/50 [==============================] - 0s 3ms/step - loss:
68848.2812
Epoch 201/500
50/50 [==============================] - 0s 3ms/step - loss:
70229.1953
Epoch 202/500
50/50 [==============================] - 0s 3ms/step - loss:
68492.6641
Epoch 203/500
50/50 [==============================] - 0s 3ms/step - loss:
61273.8008
Epoch 204/500
50/50 [==============================] - 0s 3ms/step - loss:
68777.7500
Epoch 205/500
50/50 [==============================] - 0s 3ms/step - loss:
64958.4766
Epoch 206/500
50/50 [==============================] - 0s 3ms/step - loss:
63631.3984
Epoch 207/500
50/50 [==============================] - 0s 3ms/step - loss:
61664.3555
Epoch 208/500
50/50 [==============================] - 0s 3ms/step - loss:
60782.7969
Epoch 209/500
50/50 [==============================] - 0s 3ms/step - loss:
58963.5117
Epoch 210/500
50/50 [==============================] - 0s 3ms/step - loss:
58198.0312
Epoch 211/500
50/50 [==============================] - 0s 3ms/step - loss:
56819.8945
Epoch 212/500
50/50 [==============================] - 0s 3ms/step - loss:
59131.6406
Epoch 213/500
50/50 [==============================] - 0s 3ms/step - loss:
57038.6055
Epoch 214/500
```

```
50/50 [==============================] - 0s 3ms/step - loss:
60827.0195
Epoch 215/500
50/50 [==============================] - 0s 3ms/step - loss:
58143.3945
Epoch 216/500
50/50 [==============================] - 0s 3ms/step - loss:
57932.0391
Epoch 217/500
50/50 [==============================] - 0s 3ms/step - loss:
60313.8594
Epoch 218/500
50/50 [==============================] - 0s 3ms/step - loss:
58467.3203
Epoch 219/500
50/50 [==============================] - 0s 3ms/step - loss:
62984.3867
Epoch 220/500
50/50 [==============================] - 0s 3ms/step - loss:
60519.8672
Epoch 221/500
50/50 [==============================] - 0s 3ms/step - loss:
58666.0977
Epoch 222/500
50/50 [==============================] - 0s 3ms/step - loss:
58861.0000
Epoch 223/500
50/50 [==============================] - 0s 3ms/step - loss:
61503.9883
Epoch 224/500
50/50 [==============================] - 0s 3ms/step - loss:
63713.7422
Epoch 225/500
50/50 [==============================] - 0s 3ms/step - loss:
63664.7891
Epoch 226/500
50/50 [==============================] - 0s 3ms/step - loss:
56087.8672
Epoch 227/500
50/50 [==============================] - 0s 3ms/step - loss:
90245.6484
Epoch 228/500
50/50 [==============================] - 0s 3ms/step - loss:
64300.2344
Epoch 229/500
50/50 [==============================] - 0s 3ms/step - loss:
62191.5430
Epoch 230/500
50/50 [==============================] - 0s 3ms/step - loss:
64488.8047
```

```
Epoch 231/500
50/50 [==============================] - 0s 3ms/step - loss:
61112.6953
Epoch 232/500
50/50 [==============================] - 0s 3ms/step - loss:
62740.2539
Epoch 233/500
50/50 [==============================] - 0s 3ms/step - loss:
58733.2109
Epoch 234/500
50/50 [==============================] - 0s 3ms/step - loss:
57544.7266
Epoch 235/500
50/50 [==============================] - 0s 3ms/step - loss:
57788.6094
Epoch 236/500
50/50 [==============================] - 0s 3ms/step - loss:
59000.7969
Epoch 237/500
50/50 [==============================] - 0s 3ms/step - loss:
59192.1484
Epoch 238/500
50/50 [==============================] - 0s 3ms/step - loss:
55057.9570
Epoch 239/500
50/50 [==============================] - 0s 3ms/step - loss:
55955.5156
Epoch 240/500
50/50 [==============================] - 0s 3ms/step - loss:
55525.7070
Epoch 241/500
50/50 [==============================] - 0s 3ms/step - loss:
56831.7500
Epoch 242/500
50/50 [==============================] - 0s 3ms/step - loss:
59257.9219
Epoch 243/500
50/50 [==============================] - 0s 3ms/step - loss:
54419.0859
Epoch 244/500
50/50 [==============================] - 0s 3ms/step - loss:
64931.8008
Epoch 245/500
50/50 [==============================] - 0s 3ms/step - loss:
59571.7852
Epoch 246/500
50/50 [==============================] - 0s 3ms/step - loss:
59937.0977
Epoch 247/500
50/50 [==============================] - 0s 3ms/step - loss:
```

```
58396.6094
Epoch 248/500
50/50 [==============================] - 0s 3ms/step - loss:
57908.4727
Epoch 249/500
50/50 [==============================] - 0s 3ms/step - loss:
55285.7070
Epoch 250/500
50/50 [==============================] - 0s 3ms/step - loss:
55906.8320
Epoch 251/500
50/50 [==============================] - 0s 3ms/step - loss:
52458.9492
Epoch 252/500
50/50 [==============================] - 0s 3ms/step - loss:
51149.0977
Epoch 253/500
50/50 [==============================] - 0s 3ms/step - loss:
50869.2266
Epoch 254/500
50/50 [==============================] - 0s 3ms/step - loss:
54454.4297
Epoch 255/500
50/50 [==============================] - 0s 3ms/step - loss:
52510.8711
Epoch 256/500
50/50 [==============================] - 0s 3ms/step - loss:
52144.7148
Epoch 257/500
50/50 [==============================] - 0s 3ms/step - loss:
51565.8594
Epoch 258/500
50/50 [==============================] - 0s 3ms/step - loss:
50779.6484
Epoch 259/500
50/50 [==============================] - 0s 3ms/step - loss:
52701.2188
Epoch 260/500
50/50 [==============================] - 0s 3ms/step - loss:
50140.8281
Epoch 261/500
50/50 [==============================] - 0s 3ms/step - loss:
51309.5039
Epoch 262/500
50/50 [==============================] - 0s 3ms/step - loss:
58178.9688
Epoch 263/500
50/50 [==============================] - 0s 3ms/step - loss:
51746.7539
Epoch 264/500
```

```
50/50 [==============================] - 0s 3ms/step - loss:
50348.4844
Epoch 265/500
50/50 [==============================] - 0s 3ms/step - loss:
49872.9922
Epoch 266/500
50/50 [==============================] - 0s 3ms/step - loss:
49224.8789
Epoch 267/500
50/50 [==============================] - 0s 3ms/step - loss:
55203.1055
Epoch 268/500
50/50 [==============================] - 0s 3ms/step - loss:
52665.3477
Epoch 269/500
50/50 [==============================] - 0s 3ms/step - loss:
50225.4062
Epoch 270/500
50/50 [==============================] - 0s 3ms/step - loss:
50469.9453
Epoch 271/500
50/50 [==============================] - 0s 3ms/step - loss:
51955.8438
Epoch 272/500
50/50 [==============================] - 0s 3ms/step - loss:
51334.8125
Epoch 273/500
50/50 [==============================] - 0s 3ms/step - loss:
54320.9219
Epoch 274/500
50/50 [==============================] - 0s 3ms/step - loss:
47620.9219
Epoch 275/500
50/50 [==============================] - 0s 3ms/step - loss:
49920.6250
Epoch 276/500
50/50 [==============================] - 0s 3ms/step - loss:
54241.2852
Epoch 277/500
50/50 [==============================] - 0s 3ms/step - loss:
52556.2734
Epoch 278/500
50/50 [==============================] - 0s 3ms/step - loss:
50121.9961
Epoch 279/500
50/50 [==============================] - 0s 3ms/step - loss:
52095.7109
Epoch 280/500
50/50 [==============================] - 0s 3ms/step - loss:
52615.2422
```

```
Epoch 281/500
50/50 [==============================] - 0s 3ms/step - loss:
50373.6016
Epoch 282/500
50/50 [==============================] - 0s 3ms/step - loss:
51299.1406
Epoch 283/500
50/50 [==============================] - 0s 3ms/step - loss:
50662.7148
Epoch 284/500
50/50 [==============================] - 0s 3ms/step - loss:
52836.3242
Epoch 285/500
50/50 [==============================] - 0s 3ms/step - loss:
53424.3516
Epoch 286/500
50/50 [==============================] - 0s 3ms/step - loss:
52594.0469
Epoch 287/500

50/50 [==============================] - 0s 3ms/step - loss:
56731.3594
Epoch 288/500
50/50 [==============================] - 0s 3ms/step - loss:
53536.9141
Epoch 289/500
50/50 [==============================] - 0s 3ms/step - loss:
52861.0742
Epoch 290/500
50/50 [==============================] - 0s 3ms/step - loss:
51429.3438
Epoch 291/500
50/50 [==============================] - 0s 3ms/step - loss:
51069.2266
Epoch 292/500
50/50 [==============================] - 0s 3ms/step - loss:
51850.6445
Epoch 293/500
50/50 [==============================] - 0s 3ms/step - loss:
50548.7656
Epoch 294/500
50/50 [==============================] - 0s 3ms/step - loss:
55165.5820
Epoch 295/500
50/50 [==============================] - 0s 3ms/step - loss:
52035.3594
Epoch 296/500
50/50 [==============================] - 0s 3ms/step - loss:
49966.8242
Epoch 297/500
```

```
50/50 [==============================] - 0s 3ms/step - loss:
50684.6445
Epoch 298/500
50/50 [==============================] - 0s 3ms/step - loss:
49403.1914
Epoch 299/500
50/50 [==============================] - 0s 3ms/step - loss:
47763.9023
Epoch 300/500
50/50 [==============================] - 0s 3ms/step - loss:
48224.9062
Epoch 301/500
50/50 [==============================] - 0s 3ms/step - loss:
51572.1055
Epoch 302/500
50/50 [==============================] - 0s 3ms/step - loss:
48011.6328
Epoch 303/500
50/50 [==============================] - 0s 3ms/step - loss:
46561.2852
Epoch 304/500
50/50 [==============================] - 0s 3ms/step - loss:
49773.5820
Epoch 305/500
50/50 [==============================] - 0s 3ms/step - loss:
53083.2188
Epoch 306/500
50/50 [==============================] - 0s 3ms/step - loss:
51562.1211
Epoch 307/500
50/50 [==============================] - 0s 3ms/step - loss:
49742.8125
Epoch 308/500
50/50 [==============================] - 0s 3ms/step - loss:
48635.7422
Epoch 309/500
50/50 [==============================] - 0s 3ms/step - loss:
49530.2500
Epoch 310/500
50/50 [==============================] - 0s 3ms/step - loss:
48348.0547
Epoch 311/500
50/50 [==============================] - 0s 3ms/step - loss:
48923.4727
Epoch 312/500
50/50 [==============================] - 0s 3ms/step - loss:
46835.3047
Epoch 313/500
50/50 [==============================] - 0s 3ms/step - loss:
48462.0898
```

```
Epoch 314/500
50/50 [==============================] - 0s 3ms/step - loss:
50235.6602
Epoch 315/500
50/50 [==============================] - 0s 3ms/step - loss:
45280.2188
Epoch 316/500
50/50 [==============================] - 0s 3ms/step - loss:
47158.2930
Epoch 317/500
50/50 [==============================] - 0s 3ms/step - loss:
46081.3555
Epoch 318/500
50/50 [==============================] - 0s 3ms/step - loss:
46706.2227
Epoch 319/500
50/50 [==============================] - 0s 3ms/step - loss:
45235.8633
Epoch 320/500
50/50 [==============================] - 0s 3ms/step - loss:
47879.5430
Epoch 321/500
50/50 [==============================] - 0s 3ms/step - loss:
49356.2891
Epoch 322/500
50/50 [==============================] - 0s 3ms/step - loss:
47561.1133
Epoch 323/500
50/50 [==============================] - 0s 3ms/step - loss:
50237.9531
Epoch 324/500
50/50 [==============================] - 0s 3ms/step - loss:
48332.5625
Epoch 325/500
50/50 [==============================] - 0s 3ms/step - loss:
47457.6406
Epoch 326/500
50/50 [==============================] - 0s 3ms/step - loss:
50450.6445
Epoch 327/500
50/50 [==============================] - 0s 3ms/step - loss:
58999.0078
Epoch 328/500
50/50 [==============================] - 0s 3ms/step - loss:
54757.4688
Epoch 329/500
50/50 [==============================] - 0s 3ms/step - loss:
53952.7969
Epoch 330/500
50/50 [==============================] - 0s 3ms/step - loss:
```

```
55466.8750
Epoch 331/500
50/50 [==============================] - 0s 3ms/step - loss:
53229.9023
Epoch 332/500
50/50 [==============================] - 0s 3ms/step - loss:
50377.9102
Epoch 333/500
50/50 [==============================] - 0s 3ms/step - loss:
60647.5312
Epoch 334/500
50/50 [==============================] - 0s 3ms/step - loss:
54133.2500
Epoch 335/500
50/50 [==============================] - 0s 3ms/step - loss:
52213.2656
Epoch 336/500
50/50 [==============================] - 0s 3ms/step - loss:
51226.4336
Epoch 337/500
50/50 [==============================] - 0s 3ms/step - loss:
52204.7422
Epoch 338/500
50/50 [==============================] - 0s 3ms/step - loss:
51506.4141
Epoch 339/500
50/50 [==============================] - 0s 3ms/step - loss:
51064.0391
Epoch 340/500
50/50 [==============================] - 0s 3ms/step - loss:
50199.2383
Epoch 341/500
50/50 [==============================] - 0s 3ms/step - loss:
48379.7461
Epoch 342/500
50/50 [==============================] - 0s 3ms/step - loss:
49047.9023
Epoch 343/500
50/50 [==============================] - 0s 3ms/step - loss:
50714.4688
Epoch 344/500
50/50 [==============================] - 0s 3ms/step - loss:
48661.0352
Epoch 345/500
50/50 [==============================] - 0s 3ms/step - loss:
48845.6094
Epoch 346/500
50/50 [==============================] - 0s 3ms/step - loss:
49245.3789
Epoch 347/500
```

```
50/50 [==============================] - 0s 3ms/step - loss:
50474.0117
Epoch 348/500
50/50 [==============================] - 0s 3ms/step - loss:
52211.3281
Epoch 349/500
50/50 [==============================] - 0s 3ms/step - loss:
54750.5430
Epoch 350/500
50/50 [==============================] - 0s 3ms/step - loss:
51014.5195
Epoch 351/500
50/50 [==============================] - 0s 3ms/step - loss:
49924.6367
Epoch 352/500
50/50 [==============================] - 0s 3ms/step - loss:
57038.7969
Epoch 353/500
50/50 [==============================] - 0s 3ms/step - loss:
49377.0156
Epoch 354/500
50/50 [==============================] - 0s 3ms/step - loss:
49985.6094
Epoch 355/500
50/50 [==============================] - 0s 3ms/step - loss:
49772.6250
Epoch 356/500
50/50 [==============================] - 0s 3ms/step - loss:
49183.7617
Epoch 357/500
50/50 [==============================] - 0s 3ms/step - loss:
47650.0938
Epoch 358/500
50/50 [==============================] - 0s 3ms/step - loss:
49212.8750
Epoch 359/500
50/50 [==============================] - 0s 3ms/step - loss:
47786.8555
Epoch 360/500
50/50 [==============================] - 0s 3ms/step - loss:
51487.1055
Epoch 361/500
50/50 [==============================] - 0s 3ms/step - loss:
46691.4570
Epoch 362/500
50/50 [==============================] - 0s 3ms/step - loss:
46535.8477
Epoch 363/500
50/50 [==============================] - 0s 3ms/step - loss:
47093.7852
```

```
Epoch 364/500
50/50 [==============================] - 0s 3ms/step - loss:
52634.8555
Epoch 365/500
50/50 [==============================] - 0s 3ms/step - loss:
65894.6016
Epoch 366/500
50/50 [==============================] - 0s 3ms/step - loss:
58938.6719
Epoch 367/500
50/50 [==============================] - 0s 3ms/step - loss:
54501.1641
Epoch 368/500
50/50 [==============================] - 0s 3ms/step - loss:
52201.3555
Epoch 369/500
50/50 [==============================] - 0s 3ms/step - loss:
50887.8984
Epoch 370/500
50/50 [==============================] - 0s 3ms/step - loss:
49739.9102
Epoch 371/500
50/50 [==============================] - 0s 3ms/step - loss:
48977.1836
Epoch 372/500
50/50 [==============================] - 0s 3ms/step - loss:
46608.5234
Epoch 373/500
50/50 [==============================] - 0s 3ms/step - loss:
45783.6641
Epoch 374/500
50/50 [==============================] - 0s 3ms/step - loss:
46190.1133
Epoch 375/500
50/50 [==============================] - 0s 3ms/step - loss:
44891.5430
Epoch 376/500
50/50 [==============================] - 0s 3ms/step - loss:
49186.7109
Epoch 377/500
50/50 [==============================] - 0s 3ms/step - loss:
49405.7500
Epoch 378/500
50/50 [==============================] - 0s 3ms/step - loss:
46053.7891
Epoch 379/500
50/50 [==============================] - 0s 3ms/step - loss:
46164.2383
Epoch 380/500
50/50 [==============================] - 0s 3ms/step - loss:
```

```
45039.7500
Epoch 381/500
50/50 [==============================] - 0s 3ms/step - loss:
45132.8594
Epoch 382/500

50/50 [==============================] - 0s 3ms/step - loss:
46370.6719
Epoch 383/500
50/50 [==============================] - 0s 3ms/step - loss:
44439.8320
Epoch 384/500
50/50 [==============================] - 0s 3ms/step - loss:
46508.0156
Epoch 385/500
50/50 [==============================] - 0s 3ms/step - loss:
45388.8672
Epoch 386/500
50/50 [==============================] - 0s 3ms/step - loss:
42608.9219
Epoch 387/500
50/50 [==============================] - 0s 3ms/step - loss:
43947.3477
Epoch 388/500
50/50 [==============================] - 0s 3ms/step - loss:
45329.0938
Epoch 389/500
50/50 [==============================] - 0s 3ms/step - loss:
44419.5391
Epoch 390/500
50/50 [==============================] - 0s 3ms/step - loss:
44943.0273
Epoch 391/500
50/50 [==============================] - 0s 3ms/step - loss:
46733.3477
Epoch 392/500
50/50 [==============================] - 0s 3ms/step - loss:
44704.1992
Epoch 393/500
50/50 [==============================] - 0s 3ms/step - loss:
43395.5156
Epoch 394/500
50/50 [==============================] - 0s 3ms/step - loss:
44082.1445
Epoch 395/500
50/50 [==============================] - 0s 3ms/step - loss:
43008.2422
Epoch 396/500
50/50 [==============================] - 0s 3ms/step - loss:
44937.3516
```

```
Epoch 397/500
50/50 [==============================] - 0s 3ms/step - loss:
47882.5703
Epoch 398/500
50/50 [==============================] - 0s 3ms/step - loss:
55114.7188
Epoch 399/500
50/50 [==============================] - 0s 3ms/step - loss:
46759.0664
Epoch 400/500
50/50 [==============================] - 0s 3ms/step - loss:
52759.8477
Epoch 401/500
50/50 [==============================] - 0s 3ms/step - loss:
46325.8242
Epoch 402/500
50/50 [==============================] - 0s 3ms/step - loss:
47155.5078
Epoch 403/500
50/50 [==============================] - 0s 3ms/step - loss:
50127.8164
Epoch 404/500
50/50 [==============================] - 0s 3ms/step - loss:
46226.2109
Epoch 405/500
50/50 [==============================] - 0s 3ms/step - loss:
47047.2188
Epoch 406/500
50/50 [==============================] - 0s 3ms/step - loss:
45231.1523
Epoch 407/500
50/50 [==============================] - 0s 3ms/step - loss:
44638.1641
Epoch 408/500
50/50 [==============================] - 0s 3ms/step - loss:
44930.8320
Epoch 409/500
50/50 [==============================] - 0s 3ms/step - loss:
45245.9883
Epoch 410/500
50/50 [==============================] - 0s 3ms/step - loss:
46025.2656
Epoch 411/500
50/50 [==============================] - 0s 3ms/step - loss:
46813.5586
Epoch 412/500
50/50 [==============================] - 0s 3ms/step - loss:
45196.0352
Epoch 413/500
50/50 [==============================] - 0s 3ms/step - loss:
```

```
45746.9062
Epoch 414/500
50/50 [==============================] - 0s 3ms/step - loss:
43949.9102
Epoch 415/500
50/50 [==============================] - 0s 3ms/step - loss:
46150.3828
Epoch 416/500
50/50 [==============================] - 0s 3ms/step - loss:
45282.8203
Epoch 417/500
50/50 [==============================] - 0s 3ms/step - loss:
44361.4727
Epoch 418/500
50/50 [==============================] - 0s 3ms/step - loss:
46647.7578
Epoch 419/500
50/50 [==============================] - 0s 3ms/step - loss:
81669.9844
Epoch 420/500
50/50 [==============================] - 0s 3ms/step - loss:
63071.8594
Epoch 421/500
50/50 [==============================] - 0s 3ms/step - loss:
57058.9258
Epoch 422/500
50/50 [==============================] - 0s 3ms/step - loss:
53686.0781
Epoch 423/500
50/50 [==============================] - 0s 3ms/step - loss:
51417.6562
Epoch 424/500
50/50 [==============================] - 0s 3ms/step - loss:
52102.5781
Epoch 425/500
50/50 [==============================] - 0s 3ms/step - loss:
47265.0508
Epoch 426/500
50/50 [==============================] - 0s 3ms/step - loss:
47343.4961
Epoch 427/500
50/50 [==============================] - 0s 3ms/step - loss:
45026.4531
Epoch 428/500
50/50 [==============================] - 0s 3ms/step - loss:
44805.9297
Epoch 429/500
50/50 [==============================] - 0s 3ms/step - loss:
45051.9180
Epoch 430/500
```

```
50/50 [==============================] - 0s 3ms/step - loss:
43344.5000
Epoch 431/500
50/50 [==============================] - 0s 3ms/step - loss:
43121.3047
Epoch 432/500
50/50 [==============================] - 0s 3ms/step - loss:
44002.2305
Epoch 433/500
50/50 [==============================] - 0s 3ms/step - loss:
44796.9102
Epoch 434/500
50/50 [==============================] - 0s 3ms/step - loss:
45357.2188
Epoch 435/500
50/50 [==============================] - 0s 3ms/step - loss:
44309.7852
Epoch 436/500
50/50 [==============================] - 0s 3ms/step - loss:
43601.9766
Epoch 437/500
50/50 [==============================] - 0s 3ms/step - loss:
47258.4688
Epoch 438/500
50/50 [==============================] - 0s 3ms/step - loss:
45492.3906
Epoch 439/500
50/50 [==============================] - 0s 3ms/step - loss:
46953.9609
Epoch 440/500
50/50 [==============================] - 0s 3ms/step - loss:
46821.2383
Epoch 441/500
50/50 [==============================] - 0s 3ms/step - loss:
47810.2656
Epoch 442/500
50/50 [==============================] - 0s 3ms/step - loss:
43988.8789
Epoch 443/500
50/50 [==============================] - 0s 3ms/step - loss:
44657.1367
Epoch 444/500
50/50 [==============================] - 0s 3ms/step - loss:
42066.7109
Epoch 445/500
50/50 [==============================] - 0s 3ms/step - loss:
41117.4961
Epoch 446/500
50/50 [==============================] - 0s 3ms/step - loss:
42191.3750
```

```
Epoch 447/500
50/50 [==============================] - 0s 3ms/step - loss:
41653.2109
Epoch 448/500
50/50 [==============================] - 0s 3ms/step - loss:
41152.5039
Epoch 449/500
50/50 [==============================] - 0s 3ms/step - loss:
41872.6289
Epoch 450/500
50/50 [==============================] - 0s 3ms/step - loss:
41302.3086
Epoch 451/500
50/50 [==============================] - 0s 3ms/step - loss:
40910.6211
Epoch 452/500
50/50 [==============================] - 0s 3ms/step - loss:
42001.5742
Epoch 453/500
50/50 [==============================] - 0s 3ms/step - loss:
43107.9375
Epoch 454/500
50/50 [==============================] - 0s 3ms/step - loss:
41796.4180
Epoch 455/500
50/50 [==============================] - 0s 3ms/step - loss:
43488.5352
Epoch 456/500
50/50 [==============================] - 0s 3ms/step - loss:
43360.2656
Epoch 457/500
50/50 [==============================] - 0s 3ms/step - loss:
44013.6055
Epoch 458/500
50/50 [==============================] - 0s 3ms/step - loss:
78826.6172
Epoch 459/500
50/50 [==============================] - 0s 3ms/step - loss:
72804.0781
Epoch 460/500
50/50 [==============================] - 0s 3ms/step - loss:
66247.4531
Epoch 461/500
50/50 [==============================] - 0s 3ms/step - loss:
69864.1641
Epoch 462/500
50/50 [==============================] - 0s 3ms/step - loss:
158336.5156
Epoch 463/500
50/50 [==============================] - 0s 3ms/step - loss:
```

```
89688.9219
Epoch 464/500
50/50 [==============================] - 0s 3ms/step - loss:
63290.0078
Epoch 465/500
50/50 [==============================] - 0s 3ms/step - loss:
59212.5273
Epoch 466/500
50/50 [==============================] - 0s 3ms/step - loss:
56789.0117
Epoch 467/500
50/50 [==============================] - 0s 3ms/step - loss:
54123.0859
Epoch 468/500
50/50 [==============================] - 0s 3ms/step - loss:
51161.8438
Epoch 469/500
50/50 [==============================] - 0s 3ms/step - loss:
50359.1250
Epoch 470/500
50/50 [==============================] - 0s 3ms/step - loss:
55313.0234
Epoch 471/500
50/50 [==============================] - 0s 3ms/step - loss:
52449.7070
Epoch 472/500
50/50 [==============================] - 0s 3ms/step - loss:
51058.5820
Epoch 473/500
50/50 [==============================] - 0s 3ms/step - loss:
49325.5742
Epoch 474/500
50/50 [==============================] - 0s 3ms/step - loss:
47425.1680
Epoch 475/500
50/50 [==============================] - 0s 3ms/step - loss:
49173.6680
Epoch 476/500
50/50 [==============================] - 0s 3ms/step - loss:
53192.0156
Epoch 477/500

50/50 [==============================] - 0s 3ms/step - loss:
47392.9219
Epoch 478/500
50/50 [==============================] - 0s 3ms/step - loss:
48523.0234
Epoch 479/500
50/50 [==============================] - 0s 3ms/step - loss:
47171.2109
```

```
Epoch 480/500
50/50 [==============================] - 0s 3ms/step - loss:
46567.7070
Epoch 481/500
50/50 [==============================] - 0s 3ms/step - loss:
49875.6289
Epoch 482/500
50/50 [==============================] - 0s 3ms/step - loss:
48415.7109
Epoch 483/500
50/50 [==============================] - 0s 3ms/step - loss:
47719.2031
Epoch 484/500
50/50 [==============================] - 0s 3ms/step - loss:
46446.1719
Epoch 485/500
50/50 [==============================] - 0s 3ms/step - loss:
48748.6797
Epoch 486/500
50/50 [==============================] - 0s 3ms/step - loss:
53519.6836
Epoch 487/500
50/50 [==============================] - 0s 3ms/step - loss:
49007.4453
Epoch 488/500
50/50 [==============================] - 0s 3ms/step - loss:
48671.2617
Epoch 489/500
50/50 [==============================] - 0s 3ms/step - loss:
49192.2891
Epoch 490/500
50/50 [==============================] - 0s 3ms/step - loss:
51596.4844
Epoch 491/500
50/50 [==============================] - 0s 3ms/step - loss:
51831.4375
Epoch 492/500
50/50 [==============================] - 0s 3ms/step - loss:
55554.1914
Epoch 493/500
50/50 [==============================] - 0s 3ms/step - loss:
56106.2969
Epoch 494/500
50/50 [==============================] - 0s 3ms/step - loss:
50060.0625
Epoch 495/500
50/50 [==============================] - 0s 3ms/step - loss:
45578.9766
Epoch 496/500
50/50 [==============================] - 0s 3ms/step - loss:
```

```
45210.8398
Epoch 497/500
50/50 [==============================] - 0s 3ms/step - loss:
44550.3906
Epoch 498/500
50/50 [==============================] - 0s 3ms/step - loss:
48720.3867
Epoch 499/500
50/50 [==============================] - 0s 3ms/step - loss:
46457.8906
Epoch 500/500
50/50 [==============================] - 0s 3ms/step - loss:
46670.2891

<keras.callbacks.History at 0x2582469bee0>
```

```
from numpy import array
# demonstrate prediction for next 10 days
x_input = array(all_cc[-7:])
temp_input=list(x_input)
lst_output=[]
i=0
while(i<7):

    if(len(temp_input)>7):
        x_input=array(temp_input[1:])
#         print("{} day input {}".format(i,x_input))
        #print(x_input)
        x_input = x_input.reshape((1, n_steps, n_features))
        #print(x_input)
        yhat = model.predict(x_input, verbose=0)
#         print("{} day output {}".format(i,yhat))
        temp_input.append(yhat[0][0])
        temp_input=temp_input[1:]
        #print(temp_input)
        lst_output.append(yhat[0][0])
        i=i+1
    else:
        x_input = x_input.reshape((1, n_steps, n_features))
        yhat = model.predict(x_input, verbose=0)
#         print(yhat[0])
        temp_input.append(yhat[0][0])
        lst_output.append(yhat[0][0])
        i=i+1


print(lst_output)
```

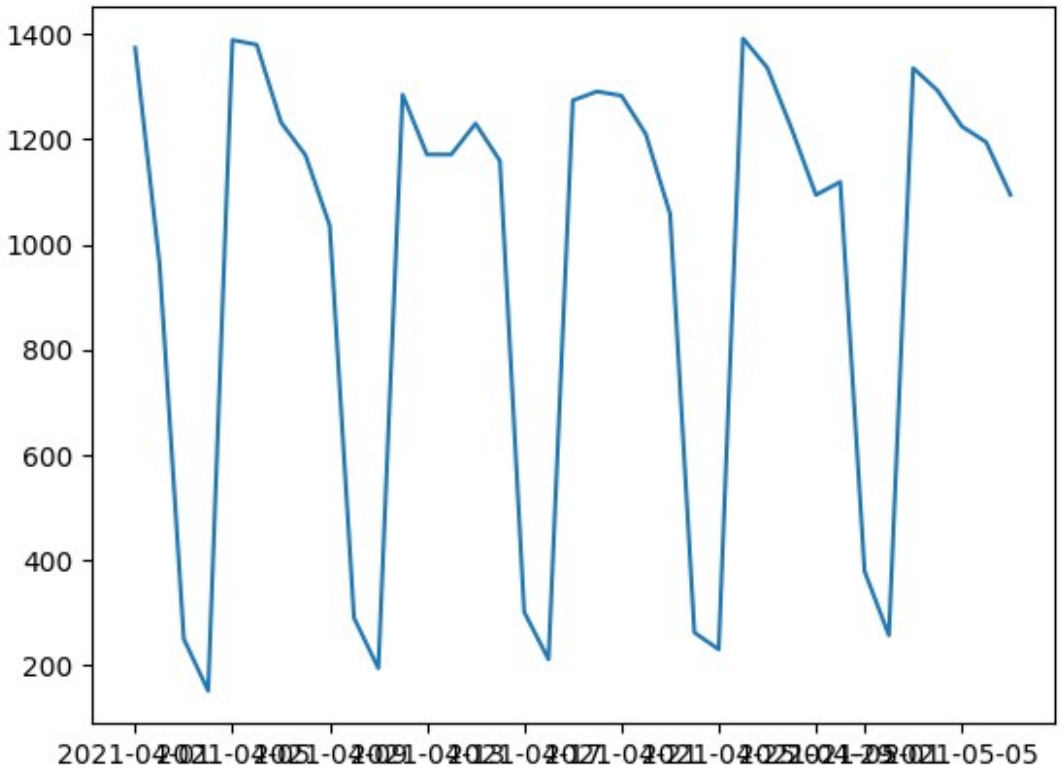
```
[377.71722, 255.75018, 1335.5135, 1292.8608, 1224.441, 1194.5858,
1094.1954]
```

```
pred_data = lstm_data.loc[ [(dt.month==4) & (dt.year==2021) for dt in
lstm_data.ds] ]
pred_data.reset_index(drop=True, inplace=True)
pred_data
```

	ds	y
0	2021-04-01	1374.0
1	2021-04-02	963.0
2	2021-04-03	249.0
3	2021-04-04	150.0
4	2021-04-05	1389.0
5	2021-04-06	1380.0
6	2021-04-07	1232.0
7	2021-04-08	1170.0
8	2021-04-09	1036.0
9	2021-04-10	289.0
10	2021-04-11	193.0
11	2021-04-12	1285.0
12	2021-04-13	1171.0
13	2021-04-14	1171.0
14	2021-04-15	1230.0
15	2021-04-16	1159.0
16	2021-04-17	300.0
17	2021-04-18	210.0
18	2021-04-19	1274.0
19	2021-04-20	1291.0
20	2021-04-21	1283.0
21	2021-04-22	1210.0
22	2021-04-23	1058.0
23	2021-04-24	261.0
24	2021-04-25	229.0
25	2021-04-26	1392.0
26	2021-04-27	1336.0
27	2021-04-28	1220.0
28	2021-04-29	1094.0
29	2021-04-30	1119.0

```
tmp = pd.concat((pred_data, pd.DataFrame({
    "ds": pd.date_range(start='5/1/2021', periods=7).strftime('%Y-%m-
%d'),
    "y": lst_output
})))
# pd.DataFrame({
#     "ds": pd.date_range(start='5/1/2021', periods=10).strftime('%Y-
%m-%d'),
#     "y": lst_output
# })

plt.plot(tmp.ds, tmp.y)
```

```
[<matplotlib.lines.Line2D at 0x258243408b0>]
```




```
tmp[-7:]
```

```
           ds            y
0  2021-05-01   377.717224
1  2021-05-02   255.750183
2  2021-05-03  1335.513550
3  2021-05-04  1292.860840
4  2021-05-05  1224.441040
5  2021-05-06  1194.585815
6  2021-05-07  1094.195435
```

```
pd.DataFrame({
    "Date": pd.date_range(start='5/1/2021',
periods=7).strftime('%m/%d/%Y'),
    "No_calls_Offered_predicted": tmp[-7:].y.to_list()
}).to_csv("week_prediction.csv", index=False)
```

```
df_17_cumm = df_17[['cc', 'rc' , 'sc',
'month']].groupby('month').sum()
```

```
years = df_not17_sorted.year.unique()
for y in years:
    df_not17_sorted_yearly = df_not17_sorted[df_not17_sorted.year ==
y]
    df_cumm = df_not17_sorted_yearly[['cc', 'rc' , 'sc',
'month']].groupby('month').sum()
#     print(df_cumm)
    df_17_cumm = pd.concat((df_17_cumm, df_cumm))

df_17_cumm
```

month	cc	rc	sc
1	48239.0	72702.0	147207.0
2	42176.0	63558.0	128753.0
3	46218.0	69673.0	140998.0
4	42362.0	63850.0	128841.0
5	45914.0	69202.0	140060.0
6	45519.0	68574.0	138623.0
7	41117.0	61970.0	125464.0
8	42484.0	63969.0	129310.0
9	39754.0	59901.0	121404.0
10	46640.0	70304.0	141922.0
11	50725.0	76379.0	154269.0
12	44771.0	67468.0	136665.0
1	48219.0	72631.0	146842.0
2	44012.0	66305.0	133977.0
3	46840.0	70598.0	142715.0
4	48348.0	72800.0	146984.0
5	48600.0	73208.0	147975.0
6	43434.0	65439.0	132629.0
7	50356.0	75806.0	153101.0
8	54320.0	81786.0	164943.0
9	48260.0	72703.0	146870.0
10	58059.0	87372.0	176099.0
11	50977.0	76781.0	155070.0
12	44124.0	66501.0	134578.0
1	46429.0	69940.0	141465.0
2	43424.0	65387.0	132108.0
3	44893.0	67669.0	136703.0
4	48295.0	72723.0	146956.0
5	50901.0	76649.0	154618.0
6	46533.0	70136.0	141740.0
7	47944.0	72248.0	146063.0
8	47310.0	71243.0	144031.0
9	45262.0	68223.0	137787.0
10	49613.0	74738.0	151231.0
11	40431.0	60984.0	123209.0
12	37935.0	57195.0	115784.0
1	44327.0	66784.0	135340.0
2	40076.0	60391.0	122104.0

```
3      38164.0  57598.0  116659.0
4      32852.0  49543.0  100659.0
5      27944.0  42197.0   86044.0
6      30685.0  46308.0   94196.0
7      30571.0  46190.0   93885.0
8      31314.0  47345.0   96343.0
9      30954.0  46713.0   94828.0
10     34179.0  51560.0  104724.0
11     30091.0  45409.0   92336.0
12     29783.0  44933.0   91702.0
1      27722.0  41872.0   85114.0
2      26341.0  39751.0   81023.0
3      32751.0  49402.0  100177.0
4      28718.0  43325.0   88096.0
```

```
lstm_data = pd.DataFrame({
    "ds": df_17_cumm.index,
    "y": df_17_cumm.cc
})
lstm_data
```

```
       ds        y
month
1       1  48239.0
2       2  42176.0
3       3  46218.0
4       4  42362.0
5       5  45914.0
6       6  45519.0
7       7  41117.0
8       8  42484.0
9       9  39754.0
10     10  46640.0
11     11  50725.0
12     12  44771.0
1       1  48219.0
2       2  44012.0
3       3  46840.0
4       4  48348.0
5       5  48600.0
6       6  43434.0
7       7  50356.0
8       8  54320.0
9       9  48260.0
10     10  58059.0
11     11  50977.0
12     12  44124.0
1       1  46429.0
2       2  43424.0
3       3  44893.0
4       4  48295.0
```

```
5       5  50901.0
6       6  46533.0
7       7  47944.0
8       8  47310.0
9       9  45262.0
10     10  49613.0
11     11  40431.0
12     12  37935.0
1       1  44327.0
2       2  40076.0
3       3  38164.0
4       4  32852.0
5       5  27944.0
6       6  30685.0
7       7  30571.0
8       8  31314.0
9       9  30954.0
10     10  34179.0
11     11  30091.0
12     12  29783.0
1       1  27722.0
2       2  26341.0
3       3  32751.0
4       4  28718.0
```

```
# define input sequence
timeseries_data = df_17_cumm.cc.to_list()
# choose a number of time steps
n_steps = 12
# split into samples
X, y = prepare_data(timeseries_data, n_steps)

# reshape from [samples, timesteps] into [samples, timesteps,
features]
n_features = 1
X = X.reshape((X.shape[0], X.shape[1], n_features))

# define model
model = Sequential()
model.add(LSTM(50, activation='relu', return_sequences=True,
input_shape=(n_steps, n_features)))
model.add(LSTM(50, activation='relu'))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
# fit model
model.fit(X, y, epochs=500, verbose=1)
```

```
Epoch 1/500
2/2 [==============================] - 2s 6ms/step - loss:
1711399168.0000
Epoch 2/500
```

```
2/2 [==============================] - 0s 5ms/step - loss:
1733785216.0000
Epoch 3/500
2/2 [==============================] - 0s 5ms/step - loss:
1751710976.0000
Epoch 4/500
2/2 [==============================] - 0s 4ms/step - loss:
1630898816.0000
Epoch 5/500
2/2 [==============================] - 0s 7ms/step - loss:
1698453888.0000
Epoch 6/500
2/2 [==============================] - 0s 7ms/step - loss:
1739772544.0000
Epoch 7/500
2/2 [==============================] - 0s 5ms/step - loss:
1713522688.0000
Epoch 8/500
2/2 [==============================] - 0s 5ms/step - loss:
1706387840.0000
Epoch 9/500
2/2 [==============================] - 0s 7ms/step - loss:
1659777280.0000
Epoch 10/500
2/2 [==============================] - 0s 7ms/step - loss:
1615126144.0000
Epoch 11/500
2/2 [==============================] - 0s 5ms/step - loss:
1606615296.0000
Epoch 12/500
2/2 [==============================] - 0s 5ms/step - loss:
1520372096.0000
Epoch 13/500
2/2 [==============================] - 0s 6ms/step - loss:
1704055936.0000
Epoch 14/500
2/2 [==============================] - 0s 6ms/step - loss:
1781650432.0000
Epoch 15/500
2/2 [==============================] - 0s 5ms/step - loss:
1763167488.0000
Epoch 16/500
2/2 [==============================] - 0s 5ms/step - loss:
1737465600.0000
Epoch 17/500
2/2 [==============================] - 0s 6ms/step - loss:
1679264512.0000
Epoch 18/500
2/2 [==============================] - 0s 6ms/step - loss:
1657613824.0000
```

```
Epoch 19/500
2/2 [==============================] - 0s 5ms/step - loss:
1617176576.0000
Epoch 20/500
2/2 [==============================] - 0s 5ms/step - loss:
1601027328.0000
Epoch 21/500
2/2 [==============================] - 0s 7ms/step - loss:
1512570496.0000
Epoch 22/500
2/2 [==============================] - 0s 5ms/step - loss:
1418384384.0000
Epoch 23/500
2/2 [==============================] - 0s 4ms/step - loss:
1544785408.0000
Epoch 24/500
2/2 [==============================] - 0s 6ms/step - loss:
1467401216.0000
Epoch 25/500
2/2 [==============================] - 0s 7ms/step - loss:
1103120896.0000
Epoch 26/500
2/2 [==============================] - 0s 6ms/step - loss:
604009920.0000
Epoch 27/500
2/2 [==============================] - 0s 4ms/step - loss:
460667648.0000
Epoch 28/500
2/2 [==============================] - 0s 6ms/step - loss:
346994080.0000
Epoch 29/500
2/2 [==============================] - 0s 6ms/step - loss:
235416880.0000
Epoch 30/500
2/2 [==============================] - 0s 5ms/step - loss:
134898272.0000
Epoch 31/500
2/2 [==============================] - 0s 6ms/step - loss:
111967768.0000
Epoch 32/500
2/2 [==============================] - 0s 7ms/step - loss:
94308400.0000
Epoch 33/500
2/2 [==============================] - 0s 6ms/step - loss:
87055600.0000
Epoch 34/500
2/2 [==============================] - 0s 5ms/step - loss:
92430704.0000
Epoch 35/500
2/2 [==============================] - 0s 5ms/step - loss:
```

```
82411504.0000
Epoch 36/500
2/2 [==============================] - 0s 6ms/step - loss:
63449652.0000
Epoch 37/500
2/2 [==============================] - 0s 4ms/step - loss:
57254472.0000
Epoch 38/500
2/2 [==============================] - 0s 5ms/step - loss:
74420992.0000
Epoch 39/500
2/2 [==============================] - 0s 5ms/step - loss:
60901864.0000
Epoch 40/500
2/2 [==============================] - 0s 5ms/step - loss:
58956576.0000
Epoch 41/500
2/2 [==============================] - 0s 4ms/step - loss:
54460512.0000
Epoch 42/500
2/2 [==============================] - 0s 6ms/step - loss:
54095500.0000
Epoch 43/500
2/2 [==============================] - 0s 4ms/step - loss:
56365024.0000
Epoch 44/500
2/2 [==============================] - 0s 4ms/step - loss:
54878604.0000
Epoch 45/500
2/2 [==============================] - 0s 6ms/step - loss:
56152536.0000
Epoch 46/500
2/2 [==============================] - 0s 6ms/step - loss:
47412568.0000
Epoch 47/500
2/2 [==============================] - 0s 5ms/step - loss:
39923320.0000
Epoch 48/500
2/2 [==============================] - 0s 5ms/step - loss:
43489632.0000
Epoch 49/500
2/2 [==============================] - 0s 6ms/step - loss:
56389784.0000
Epoch 50/500
2/2 [==============================] - 0s 5ms/step - loss:
69883688.0000
Epoch 51/500
2/2 [==============================] - 0s 6ms/step - loss:
91846192.0000
Epoch 52/500
```

```
2/2 [==============================] - 0s 6ms/step - loss:
81893464.0000
Epoch 53/500
2/2 [==============================] - 0s 7ms/step - loss:
79905248.0000
Epoch 54/500
2/2 [==============================] - 0s 7ms/step - loss:
58530240.0000
Epoch 55/500
2/2 [==============================] - 0s 5ms/step - loss:
55558136.0000
Epoch 56/500
2/2 [==============================] - 0s 5ms/step - loss:
76524952.0000
Epoch 57/500
2/2 [==============================] - 0s 6ms/step - loss:
73806288.0000
Epoch 58/500
2/2 [==============================] - 0s 7ms/step - loss:
83651216.0000
Epoch 59/500
2/2 [==============================] - 0s 6ms/step - loss:
57914784.0000
Epoch 60/500
2/2 [==============================] - 0s 5ms/step - loss:
52215360.0000
Epoch 61/500
2/2 [==============================] - 0s 6ms/step - loss:
54818964.0000
Epoch 62/500
2/2 [==============================] - 0s 6ms/step - loss:
56918296.0000
Epoch 63/500
2/2 [==============================] - 0s 5ms/step - loss:
61419040.0000
Epoch 64/500
2/2 [==============================] - 0s 5ms/step - loss:
62542900.0000
Epoch 65/500
2/2 [==============================] - 0s 8ms/step - loss:
55681876.0000
Epoch 66/500
2/2 [==============================] - 0s 8ms/step - loss:
51960264.0000
Epoch 67/500
2/2 [==============================] - 0s 5ms/step - loss:
49496776.0000
Epoch 68/500
2/2 [==============================] - 0s 5ms/step - loss:
46800624.0000
```

```
Epoch 69/500
2/2 [==============================] - 0s 5ms/step - loss:
45799792.0000
Epoch 70/500
2/2 [==============================] - 0s 7ms/step - loss:
51362756.0000
Epoch 71/500
2/2 [==============================] - 0s 8ms/step - loss:
43207412.0000
Epoch 72/500
2/2 [==============================] - 0s 6ms/step - loss:
49633752.0000
Epoch 73/500
2/2 [==============================] - 0s 5ms/step - loss:
41920480.0000
Epoch 74/500
2/2 [==============================] - 0s 6ms/step - loss:
45115092.0000
Epoch 75/500
2/2 [==============================] - 0s 7ms/step - loss:
50119104.0000
Epoch 76/500
2/2 [==============================] - 0s 7ms/step - loss:
61677120.0000
Epoch 77/500
2/2 [==============================] - 0s 5ms/step - loss:
56030632.0000
Epoch 78/500
2/2 [==============================] - 0s 5ms/step - loss:
62001920.0000
Epoch 79/500
2/2 [==============================] - 0s 7ms/step - loss:
70496744.0000
Epoch 80/500
2/2 [==============================] - 0s 7ms/step - loss:
74757336.0000
Epoch 81/500
2/2 [==============================] - 0s 5ms/step - loss:
76514920.0000
Epoch 82/500
2/2 [==============================] - 0s 4ms/step - loss:
76084720.0000
Epoch 83/500
2/2 [==============================] - 0s 7ms/step - loss:
75197648.0000
Epoch 84/500
2/2 [==============================] - 0s 7ms/step - loss:
72916624.0000
Epoch 85/500
2/2 [==============================] - 0s 6ms/step - loss:
```

```
69426920.0000
Epoch 86/500
2/2 [==============================] - 0s 4ms/step - loss:
67598296.0000
Epoch 87/500
2/2 [==============================] - 0s 6ms/step - loss:
61422632.0000
Epoch 88/500
2/2 [==============================] - 0s 6ms/step - loss:
60241160.0000
Epoch 89/500
2/2 [==============================] - 0s 4ms/step - loss:
69370600.0000
Epoch 90/500
2/2 [==============================] - 0s 6ms/step - loss:
61073208.0000
Epoch 91/500
2/2 [==============================] - 0s 8ms/step - loss:
59684076.0000
Epoch 92/500
2/2 [==============================] - 0s 6ms/step - loss:
62912160.0000
Epoch 93/500
2/2 [==============================] - 0s 5ms/step - loss:
59243712.0000
Epoch 94/500
2/2 [==============================] - 0s 5ms/step - loss:
56298424.0000
Epoch 95/500

2/2 [==============================] - 0s 6ms/step - loss:
54145240.0000
Epoch 96/500
2/2 [==============================] - 0s 5ms/step - loss:
54033600.0000
Epoch 97/500
2/2 [==============================] - 0s 5ms/step - loss:
51446060.0000
Epoch 98/500
2/2 [==============================] - 0s 6ms/step - loss:
50044096.0000
Epoch 99/500
2/2 [==============================] - 0s 7ms/step - loss:
50847176.0000
Epoch 100/500
2/2 [==============================] - 0s 7ms/step - loss:
50968440.0000
Epoch 101/500
2/2 [==============================] - 0s 6ms/step - loss:
50552676.0000
```

```
Epoch 102/500
2/2 [==============================] - 0s 5ms/step - loss:
78714296.0000
Epoch 103/500
2/2 [==============================] - 0s 6ms/step - loss:
63657504.0000
Epoch 104/500
2/2 [==============================] - 0s 7ms/step - loss:
40070064.0000
Epoch 105/500
2/2 [==============================] - 0s 5ms/step - loss:
36426584.0000
Epoch 106/500
2/2 [==============================] - 0s 5ms/step - loss:
48976960.0000
Epoch 107/500
2/2 [==============================] - 0s 6ms/step - loss:
55400244.0000
Epoch 108/500
2/2 [==============================] - 0s 6ms/step - loss:
48157144.0000
Epoch 109/500
2/2 [==============================] - 0s 7ms/step - loss:
40440816.0000
Epoch 110/500
2/2 [==============================] - 0s 5ms/step - loss:
38757808.0000
Epoch 111/500
2/2 [==============================] - 0s 5ms/step - loss:
40548392.0000
Epoch 112/500
2/2 [==============================] - 0s 6ms/step - loss:
48155304.0000
Epoch 113/500
2/2 [==============================] - 0s 7ms/step - loss:
44133484.0000
Epoch 114/500
2/2 [==============================] - 0s 6ms/step - loss:
40120608.0000
Epoch 115/500
2/2 [==============================] - 0s 4ms/step - loss:
37735528.0000
Epoch 116/500
2/2 [==============================] - 0s 5ms/step - loss:
42221300.0000
Epoch 117/500
2/2 [==============================] - 0s 8ms/step - loss:
41872404.0000
Epoch 118/500
2/2 [==============================] - 0s 8ms/step - loss:
```

```
44152104.0000
Epoch 119/500
2/2 [==============================] - 0s 4ms/step - loss:
39156808.0000
Epoch 120/500
2/2 [==============================] - 0s 5ms/step - loss:
36378856.0000
Epoch 121/500
2/2 [==============================] - 0s 6ms/step - loss:
37823748.0000
Epoch 122/500
2/2 [==============================] - 0s 7ms/step - loss:
39561888.0000
Epoch 123/500
2/2 [==============================] - 0s 5ms/step - loss:
39030428.0000
Epoch 124/500
2/2 [==============================] - 0s 5ms/step - loss:
37321928.0000
Epoch 125/500
2/2 [==============================] - 0s 8ms/step - loss:
35164764.0000
Epoch 126/500
2/2 [==============================] - 0s 4ms/step - loss:
33436100.0000
Epoch 127/500
2/2 [==============================] - 0s 5ms/step - loss:
51067796.0000
Epoch 128/500
2/2 [==============================] - 0s 6ms/step - loss:
47314024.0000
Epoch 129/500
2/2 [==============================] - 0s 6ms/step - loss:
42698568.0000
Epoch 130/500
2/2 [==============================] - 0s 5ms/step - loss:
41492024.0000
Epoch 131/500
2/2 [==============================] - 0s 5ms/step - loss:
42718416.0000
Epoch 132/500
2/2 [==============================] - 0s 6ms/step - loss:
43472960.0000
Epoch 133/500
2/2 [==============================] - 0s 6ms/step - loss:
41246580.0000
Epoch 134/500
2/2 [==============================] - 0s 5ms/step - loss:
38423392.0000
Epoch 135/500
```

```
2/2 [==============================] - 0s 5ms/step - loss:
38397572.0000
Epoch 136/500
2/2 [==============================] - 0s 8ms/step - loss:
39158552.0000
Epoch 137/500
2/2 [==============================] - 0s 6ms/step - loss:
38885776.0000
Epoch 138/500
2/2 [==============================] - 0s 4ms/step - loss:
39120952.0000
Epoch 139/500
2/2 [==============================] - 0s 4ms/step - loss:
39555184.0000
Epoch 140/500
2/2 [==============================] - 0s 6ms/step - loss:
40323252.0000
Epoch 141/500
2/2 [==============================] - 0s 5ms/step - loss:
41742008.0000
Epoch 142/500
2/2 [==============================] - 0s 4ms/step - loss:
41583352.0000
Epoch 143/500
2/2 [==============================] - 0s 7ms/step - loss:
41058572.0000
Epoch 144/500
2/2 [==============================] - 0s 11ms/step - loss:
41319784.0000
Epoch 145/500
2/2 [==============================] - 0s 8ms/step - loss:
42103720.0000
Epoch 146/500
2/2 [==============================] - 0s 6ms/step - loss:
42681228.0000
Epoch 147/500
2/2 [==============================] - 0s 5ms/step - loss:
42296468.0000
Epoch 148/500
2/2 [==============================] - 0s 5ms/step - loss:
40730324.0000
Epoch 149/500
2/2 [==============================] - 0s 5ms/step - loss:
40051128.0000
Epoch 150/500
2/2 [==============================] - 0s 8ms/step - loss:
41520812.0000
Epoch 151/500
2/2 [==============================] - 0s 9ms/step - loss:
41103924.0000
```

```
Epoch 152/500
2/2 [==============================] - 0s 7ms/step - loss:
41183288.0000
Epoch 153/500
2/2 [==============================] - 0s 4ms/step - loss:
40552532.0000
Epoch 154/500
2/2 [==============================] - 0s 5ms/step - loss:
40456504.0000
Epoch 155/500
2/2 [==============================] - 0s 7ms/step - loss:
40312152.0000
Epoch 156/500
2/2 [==============================] - 0s 6ms/step - loss:
40281208.0000
Epoch 157/500
2/2 [==============================] - 0s 6ms/step - loss:
40483384.0000
Epoch 158/500
2/2 [==============================] - 0s 6ms/step - loss:
41052700.0000
Epoch 159/500
2/2 [==============================] - 0s 6ms/step - loss:
41789792.0000
Epoch 160/500
2/2 [==============================] - 0s 7ms/step - loss:
41606492.0000
Epoch 161/500
2/2 [==============================] - 0s 5ms/step - loss:
41051336.0000
Epoch 162/500
2/2 [==============================] - 0s 5ms/step - loss:
39782336.0000
Epoch 163/500
2/2 [==============================] - 0s 5ms/step - loss:
39334124.0000
Epoch 164/500
2/2 [==============================] - 0s 7ms/step - loss:
39628792.0000
Epoch 165/500
2/2 [==============================] - 0s 5ms/step - loss:
41638472.0000
Epoch 166/500
2/2 [==============================] - 0s 5ms/step - loss:
44383812.0000
Epoch 167/500
2/2 [==============================] - 0s 5ms/step - loss:
43886920.0000
Epoch 168/500
2/2 [==============================] - 0s 7ms/step - loss:
```

```
42095888.0000
Epoch 169/500
2/2 [==============================] - 0s 8ms/step - loss:
40140568.0000
Epoch 170/500
2/2 [==============================] - 0s 5ms/step - loss:
39329644.0000
Epoch 171/500
2/2 [==============================] - 0s 5ms/step - loss:
39331596.0000
Epoch 172/500
2/2 [==============================] - 0s 6ms/step - loss:
42119624.0000
Epoch 173/500
2/2 [==============================] - 0s 8ms/step - loss:
41411356.0000
Epoch 174/500
2/2 [==============================] - 0s 5ms/step - loss:
41826136.0000
Epoch 175/500
2/2 [==============================] - 0s 5ms/step - loss:
42900264.0000
Epoch 176/500
2/2 [==============================] - 0s 6ms/step - loss:
41696300.0000
Epoch 177/500
2/2 [==============================] - 0s 7ms/step - loss:
42061800.0000
Epoch 178/500
2/2 [==============================] - 0s 5ms/step - loss:
42698436.0000
Epoch 179/500
2/2 [==============================] - 0s 5ms/step - loss:
43486732.0000
Epoch 180/500
2/2 [==============================] - 0s 7ms/step - loss:
42652072.0000
Epoch 181/500
2/2 [==============================] - 0s 7ms/step - loss:
41528012.0000
Epoch 182/500
2/2 [==============================] - 0s 4ms/step - loss:
40757656.0000
Epoch 183/500
2/2 [==============================] - 0s 4ms/step - loss:
38814268.0000
Epoch 184/500
2/2 [==============================] - 0s 6ms/step - loss:
39063060.0000
Epoch 185/500
```

```
2/2 [==============================] - 0s 7ms/step - loss:
37334248.0000
Epoch 186/500
2/2 [==============================] - 0s 5ms/step - loss:
37797332.0000
Epoch 187/500
2/2 [==============================] - 0s 5ms/step - loss:
38377868.0000
Epoch 188/500
2/2 [==============================] - 0s 5ms/step - loss:
44277464.0000
Epoch 189/500

2/2 [==============================] - 0s 7ms/step - loss:
47277444.0000
Epoch 190/500
2/2 [==============================] - 0s 5ms/step - loss:
37843008.0000
Epoch 191/500
2/2 [==============================] - 0s 5ms/step - loss:
39021112.0000
Epoch 192/500
2/2 [==============================] - 0s 7ms/step - loss:
37581944.0000
Epoch 193/500
2/2 [==============================] - 0s 6ms/step - loss:
32806118.0000
Epoch 194/500
2/2 [==============================] - 0s 5ms/step - loss:
33484246.0000
Epoch 195/500
2/2 [==============================] - 0s 6ms/step - loss:
25744468.0000
Epoch 196/500
2/2 [==============================] - 0s 6ms/step - loss:
25867676.0000
Epoch 197/500
2/2 [==============================] - 0s 7ms/step - loss:
29323244.0000
Epoch 198/500
2/2 [==============================] - 0s 5ms/step - loss:
31888918.0000
Epoch 199/500
2/2 [==============================] - 0s 6ms/step - loss:
29079402.0000
Epoch 200/500
2/2 [==============================] - 0s 7ms/step - loss:
30805908.0000
Epoch 201/500
2/2 [==============================] - 0s 8ms/step - loss:
```

```
30575156.0000
Epoch 202/500
2/2 [==============================] - 0s 4ms/step - loss:
28914628.0000
Epoch 203/500
2/2 [==============================] - 0s 5ms/step - loss:
32545498.0000
Epoch 204/500
2/2 [==============================] - 0s 7ms/step - loss:
30923508.0000
Epoch 205/500
2/2 [==============================] - 0s 5ms/step - loss:
26985158.0000
Epoch 206/500
2/2 [==============================] - 0s 5ms/step - loss:
23873220.0000
Epoch 207/500
2/2 [==============================] - 0s 6ms/step - loss:
27718566.0000
Epoch 208/500
2/2 [==============================] - 0s 8ms/step - loss:
33284288.0000
Epoch 209/500
2/2 [==============================] - 0s 4ms/step - loss:
43905468.0000
Epoch 210/500
2/2 [==============================] - 0s 5ms/step - loss:
63465976.0000
Epoch 211/500
2/2 [==============================] - 0s 6ms/step - loss:
58564756.0000
Epoch 212/500
2/2 [==============================] - 0s 6ms/step - loss:
50805508.0000
Epoch 213/500
2/2 [==============================] - 0s 4ms/step - loss:
50725192.0000
Epoch 214/500
2/2 [==============================] - 0s 5ms/step - loss:
52229032.0000
Epoch 215/500
2/2 [==============================] - 0s 6ms/step - loss:
51406448.0000
Epoch 216/500
2/2 [==============================] - 0s 6ms/step - loss:
48509144.0000
Epoch 217/500
2/2 [==============================] - 0s 5ms/step - loss:
45750176.0000
Epoch 218/500
```

```
2/2 [==============================] - 0s 5ms/step - loss:
45089388.0000
Epoch 219/500
2/2 [==============================] - 0s 7ms/step - loss:
44796596.0000
Epoch 220/500
2/2 [==============================] - 0s 5ms/step - loss:
44385324.0000
Epoch 221/500
2/2 [==============================] - 0s 5ms/step - loss:
44359800.0000
Epoch 222/500
2/2 [==============================] - 0s 7ms/step - loss:
43717080.0000
Epoch 223/500
2/2 [==============================] - 0s 6ms/step - loss:
42732328.0000
Epoch 224/500
2/2 [==============================] - 0s 4ms/step - loss:
42152988.0000
Epoch 225/500
2/2 [==============================] - 0s 5ms/step - loss:
42412196.0000
Epoch 226/500
2/2 [==============================] - 0s 8ms/step - loss:
42223984.0000
Epoch 227/500
2/2 [==============================] - 0s 5ms/step - loss:
41690036.0000
Epoch 228/500
2/2 [==============================] - 0s 4ms/step - loss:
40761432.0000
Epoch 229/500
2/2 [==============================] - 0s 6ms/step - loss:
41484828.0000
Epoch 230/500
2/2 [==============================] - 0s 7ms/step - loss:
40635112.0000
Epoch 231/500
2/2 [==============================] - 0s 6ms/step - loss:
39738120.0000
Epoch 232/500
2/2 [==============================] - 0s 5ms/step - loss:
39518936.0000
Epoch 233/500
2/2 [==============================] - 0s 6ms/step - loss:
38828168.0000
Epoch 234/500
2/2 [==============================] - 0s 6ms/step - loss:
38546152.0000
```

```
Epoch 235/500
2/2 [==============================] - 0s 5ms/step - loss:
39586444.0000
Epoch 236/500
2/2 [==============================] - 0s 5ms/step - loss:
39196684.0000
Epoch 237/500
2/2 [==============================] - 0s 6ms/step - loss:
38308708.0000
Epoch 238/500
2/2 [==============================] - 0s 5ms/step - loss:
37609196.0000
Epoch 239/500
2/2 [==============================] - 0s 4ms/step - loss:
33248548.0000
Epoch 240/500
2/2 [==============================] - 0s 5ms/step - loss:
34794904.0000
Epoch 241/500
2/2 [==============================] - 0s 7ms/step - loss:
35752288.0000
Epoch 242/500
2/2 [==============================] - 0s 8ms/step - loss:
34675672.0000
Epoch 243/500
2/2 [==============================] - 0s 6ms/step - loss:
35056484.0000
Epoch 244/500
2/2 [==============================] - 0s 4ms/step - loss:
40715776.0000
Epoch 245/500
2/2 [==============================] - 0s 5ms/step - loss:
39282368.0000
Epoch 246/500
2/2 [==============================] - 0s 6ms/step - loss:
38469796.0000
Epoch 247/500
2/2 [==============================] - 0s 4ms/step - loss:
37788712.0000
Epoch 248/500
2/2 [==============================] - 0s 5ms/step - loss:
38992840.0000
Epoch 249/500
2/2 [==============================] - 0s 6ms/step - loss:
39411352.0000
Epoch 250/500
2/2 [==============================] - 0s 5ms/step - loss:
38807980.0000
Epoch 251/500
2/2 [==============================] - 0s 5ms/step - loss:
```

```
36847020.0000
Epoch 252/500
2/2 [==============================] - 0s 6ms/step - loss:
32613772.0000
Epoch 253/500
2/2 [==============================] - 0s 7ms/step - loss:
36287592.0000
Epoch 254/500
2/2 [==============================] - 0s 6ms/step - loss:
39579216.0000
Epoch 255/500
2/2 [==============================] - 0s 6ms/step - loss:
38800408.0000
Epoch 256/500
2/2 [==============================] - 0s 5ms/step - loss:
36279764.0000
Epoch 257/500
2/2 [==============================] - 0s 6ms/step - loss:
34981996.0000
Epoch 258/500
2/2 [==============================] - 0s 5ms/step - loss:
37854596.0000
Epoch 259/500
2/2 [==============================] - 0s 5ms/step - loss:
36531892.0000
Epoch 260/500
2/2 [==============================] - 0s 7ms/step - loss:
35524776.0000
Epoch 261/500
2/2 [==============================] - 0s 4ms/step - loss:
47908064.0000
Epoch 262/500
2/2 [==============================] - 0s 4ms/step - loss:
47786296.0000
Epoch 263/500
2/2 [==============================] - 0s 6ms/step - loss:
41735168.0000
Epoch 264/500
2/2 [==============================] - 0s 6ms/step - loss:
36719208.0000
Epoch 265/500
2/2 [==============================] - 0s 4ms/step - loss:
36421432.0000
Epoch 266/500
2/2 [==============================] - 0s 6ms/step - loss:
43573288.0000
Epoch 267/500
2/2 [==============================] - 0s 7ms/step - loss:
43408748.0000
Epoch 268/500
```

```
2/2 [==============================] - 0s 4ms/step - loss:
39619108.0000
Epoch 269/500
2/2 [==============================] - 0s 5ms/step - loss:
36456864.0000
Epoch 270/500
2/2 [==============================] - 0s 6ms/step - loss:
36166304.0000
Epoch 271/500
2/2 [==============================] - 0s 5ms/step - loss:
38173172.0000
Epoch 272/500
2/2 [==============================] - 0s 5ms/step - loss:
39252940.0000
Epoch 273/500
2/2 [==============================] - 0s 6ms/step - loss:
38315416.0000
Epoch 274/500
2/2 [==============================] - 0s 6ms/step - loss:
35328932.0000
Epoch 275/500
2/2 [==============================] - 0s 5ms/step - loss:
31722310.0000
Epoch 276/500
2/2 [==============================] - 0s 5ms/step - loss:
34322236.0000
Epoch 277/500
2/2 [==============================] - 0s 6ms/step - loss:
34624992.0000
Epoch 278/500
2/2 [==============================] - 0s 6ms/step - loss:
33693368.0000
Epoch 279/500
2/2 [==============================] - 0s 5ms/step - loss:
33820904.0000
Epoch 280/500
2/2 [==============================] - 0s 6ms/step - loss:
35001416.0000
Epoch 281/500
2/2 [==============================] - 0s 6ms/step - loss:
35316112.0000
Epoch 282/500
2/2 [==============================] - 0s 7ms/step - loss:
34716564.0000
Epoch 283/500

2/2 [==============================] - 0s 6ms/step - loss:
35015628.0000
Epoch 284/500
2/2 [==============================] - 0s 5ms/step - loss:
```

```
34945284.0000
Epoch 285/500
2/2 [==============================] - 0s 5ms/step - loss:
35351360.0000
Epoch 286/500
2/2 [==============================] - 0s 7ms/step - loss:
35358164.0000
Epoch 287/500
2/2 [==============================] - 0s 4ms/step - loss:
34867624.0000
Epoch 288/500
2/2 [==============================] - 0s 5ms/step - loss:
35386568.0000
Epoch 289/500
2/2 [==============================] - 0s 5ms/step - loss:
35709736.0000
Epoch 290/500
2/2 [==============================] - 0s 6ms/step - loss:
35737032.0000
Epoch 291/500
2/2 [==============================] - 0s 5ms/step - loss:
34921608.0000
Epoch 292/500
2/2 [==============================] - 0s 5ms/step - loss:
34773888.0000
Epoch 293/500
2/2 [==============================] - 0s 6ms/step - loss:
34656984.0000
Epoch 294/500
2/2 [==============================] - 0s 5ms/step - loss:
34686784.0000
Epoch 295/500
2/2 [==============================] - 0s 5ms/step - loss:
34630200.0000
Epoch 296/500
2/2 [==============================] - 0s 6ms/step - loss:
34480056.0000
Epoch 297/500
2/2 [==============================] - 0s 6ms/step - loss:
33346624.0000
Epoch 298/500
2/2 [==============================] - 0s 5ms/step - loss:
32072364.0000
Epoch 299/500
2/2 [==============================] - 0s 6ms/step - loss:
31894678.0000
Epoch 300/500
2/2 [==============================] - 0s 7ms/step - loss:
31577328.0000
Epoch 301/500
```

```
2/2 [==============================] - 0s 5ms/step - loss:
31168980.0000
Epoch 302/500
2/2 [==============================] - 0s 5ms/step - loss:
30728166.0000
Epoch 303/500
2/2 [==============================] - 0s 6ms/step - loss:
30631636.0000
Epoch 304/500
2/2 [==============================] - 0s 6ms/step - loss:
30352700.0000
Epoch 305/500
2/2 [==============================] - 0s 4ms/step - loss:
30034540.0000
Epoch 306/500
2/2 [==============================] - 0s 6ms/step - loss:
29938964.0000
Epoch 307/500
2/2 [==============================] - 0s 7ms/step - loss:
29438804.0000
Epoch 308/500
2/2 [==============================] - 0s 5ms/step - loss:
29415708.0000
Epoch 309/500
2/2 [==============================] - 0s 5ms/step - loss:
33756536.0000
Epoch 310/500
2/2 [==============================] - 0s 7ms/step - loss:
36319608.0000
Epoch 311/500
2/2 [==============================] - 0s 6ms/step - loss:
33271236.0000
Epoch 312/500
2/2 [==============================] - 0s 5ms/step - loss:
33893776.0000
Epoch 313/500
2/2 [==============================] - 0s 5ms/step - loss:
36994152.0000
Epoch 314/500
2/2 [==============================] - 0s 7ms/step - loss:
34698112.0000
Epoch 315/500
2/2 [==============================] - 0s 6ms/step - loss:
33618560.0000
Epoch 316/500
2/2 [==============================] - 0s 4ms/step - loss:
37460628.0000
Epoch 317/500
2/2 [==============================] - 0s 6ms/step - loss:
37590376.0000
```

```
Epoch 318/500
2/2 [==============================] - 0s 8ms/step - loss:
33295092.0000
Epoch 319/500
2/2 [==============================] - 0s 5ms/step - loss:
35530120.0000
Epoch 320/500
2/2 [==============================] - 0s 4ms/step - loss:
40037208.0000
Epoch 321/500
2/2 [==============================] - 0s 6ms/step - loss:
36252268.0000
Epoch 322/500
2/2 [==============================] - 0s 6ms/step - loss:
33096682.0000
Epoch 323/500
2/2 [==============================] - 0s 5ms/step - loss:
36918120.0000
Epoch 324/500
2/2 [==============================] - 0s 6ms/step - loss:
38491736.0000
Epoch 325/500
2/2 [==============================] - 0s 7ms/step - loss:
35588952.0000
Epoch 326/500
2/2 [==============================] - 0s 6ms/step - loss:
33044908.0000
Epoch 327/500
2/2 [==============================] - 0s 4ms/step - loss:
33682272.0000
Epoch 328/500
2/2 [==============================] - 0s 8ms/step - loss:
35258792.0000
Epoch 329/500
2/2 [==============================] - 0s 8ms/step - loss:
34670316.0000
Epoch 330/500
2/2 [==============================] - 0s 11ms/step - loss:
33996364.0000
Epoch 331/500
2/2 [==============================] - 0s 5ms/step - loss:
34415208.0000
Epoch 332/500
2/2 [==============================] - 0s 6ms/step - loss:
35738408.0000
Epoch 333/500
2/2 [==============================] - 0s 5ms/step - loss:
35802936.0000
Epoch 334/500
2/2 [==============================] - 0s 6ms/step - loss:
```

```
33922420.0000
Epoch 335/500
2/2 [==============================] - 0s 6ms/step - loss:
35723416.0000
Epoch 336/500
2/2 [==============================] - 0s 6ms/step - loss:
39632472.0000
Epoch 337/500
2/2 [==============================] - 0s 4ms/step - loss:
37549204.0000
Epoch 338/500
2/2 [==============================] - 0s 6ms/step - loss:
35039732.0000
Epoch 339/500
2/2 [==============================] - 0s 6ms/step - loss:
37728652.0000
Epoch 340/500
2/2 [==============================] - 0s 5ms/step - loss:
37448836.0000
Epoch 341/500
2/2 [==============================] - 0s 5ms/step - loss:
34644040.0000
Epoch 342/500
2/2 [==============================] - 0s 6ms/step - loss:
35116396.0000
Epoch 343/500
2/2 [==============================] - 0s 7ms/step - loss:
36219336.0000
Epoch 344/500
2/2 [==============================] - 0s 5ms/step - loss:
35008916.0000
Epoch 345/500
2/2 [==============================] - 0s 5ms/step - loss:
35273236.0000
Epoch 346/500
2/2 [==============================] - 0s 7ms/step - loss:
35186176.0000
Epoch 347/500
2/2 [==============================] - 0s 6ms/step - loss:
35351420.0000
Epoch 348/500
2/2 [==============================] - 0s 5ms/step - loss:
34759268.0000
Epoch 349/500
2/2 [==============================] - 0s 5ms/step - loss:
33922320.0000
Epoch 350/500
2/2 [==============================] - 0s 7ms/step - loss:
34643688.0000
Epoch 351/500
```

```
2/2 [==============================] - 0s 5ms/step - loss:
34838316.0000
Epoch 352/500
2/2 [==============================] - 0s 5ms/step - loss:
34552964.0000
Epoch 353/500
2/2 [==============================] - 0s 5ms/step - loss:
34216508.0000
Epoch 354/500
2/2 [==============================] - 0s 7ms/step - loss:
33957492.0000
Epoch 355/500
2/2 [==============================] - 0s 5ms/step - loss:
34081784.0000
Epoch 356/500
2/2 [==============================] - 0s 5ms/step - loss:
35424480.0000
Epoch 357/500
2/2 [==============================] - 0s 6ms/step - loss:
34434724.0000
Epoch 358/500
2/2 [==============================] - 0s 5ms/step - loss:
33619964.0000
Epoch 359/500
2/2 [==============================] - 0s 5ms/step - loss:
34657588.0000
Epoch 360/500
2/2 [==============================] - 0s 6ms/step - loss:
34854776.0000
Epoch 361/500
2/2 [==============================] - 0s 5ms/step - loss:
33998384.0000
Epoch 362/500
2/2 [==============================] - 0s 5ms/step - loss:
33779240.0000
Epoch 363/500
2/2 [==============================] - 0s 5ms/step - loss:
33638216.0000
Epoch 364/500
2/2 [==============================] - 0s 6ms/step - loss:
33244042.0000
Epoch 365/500
2/2 [==============================] - 0s 5ms/step - loss:
33705976.0000
Epoch 366/500
2/2 [==============================] - 0s 5ms/step - loss:
34945220.0000
Epoch 367/500
2/2 [==============================] - 0s 7ms/step - loss:
34899612.0000
```

```
Epoch 368/500
2/2 [==============================] - 0s 5ms/step - loss:
33125888.0000
Epoch 369/500
2/2 [==============================] - 0s 4ms/step - loss:
33906136.0000
Epoch 370/500
2/2 [==============================] - 0s 5ms/step - loss:
35609048.0000
Epoch 371/500
2/2 [==============================] - 0s 6ms/step - loss:
35310904.0000
Epoch 372/500
2/2 [==============================] - 0s 4ms/step - loss:
33260394.0000
Epoch 373/500
2/2 [==============================] - 0s 5ms/step - loss:
33013850.0000
Epoch 374/500
2/2 [==============================] - 0s 6ms/step - loss:
34247356.0000
Epoch 375/500
2/2 [==============================] - 0s 5ms/step - loss:
35622184.0000
Epoch 376/500
2/2 [==============================] - 0s 4ms/step - loss:
35845588.0000
Epoch 377/500

2/2 [==============================] - 0s 7ms/step - loss:
34633456.0000
Epoch 378/500
2/2 [==============================] - 0s 5ms/step - loss:
33954496.0000
Epoch 379/500
2/2 [==============================] - 0s 4ms/step - loss:
33368314.0000
Epoch 380/500
2/2 [==============================] - 0s 5ms/step - loss:
32876852.0000
Epoch 381/500
2/2 [==============================] - 0s 6ms/step - loss:
33042524.0000
Epoch 382/500
2/2 [==============================] - 0s 5ms/step - loss:
33512244.0000
Epoch 383/500
2/2 [==============================] - 0s 5ms/step - loss:
33662200.0000
Epoch 384/500
```

```
2/2 [==============================] - 0s 6ms/step - loss:
33235028.0000
Epoch 385/500
2/2 [==============================] - 0s 7ms/step - loss:
32688592.0000
Epoch 386/500
2/2 [==============================] - 0s 6ms/step - loss:
32430564.0000
Epoch 387/500
2/2 [==============================] - 0s 5ms/step - loss:
32645952.0000
Epoch 388/500
2/2 [==============================] - 0s 6ms/step - loss:
32828756.0000
Epoch 389/500
2/2 [==============================] - 0s 6ms/step - loss:
32106662.0000
Epoch 390/500
2/2 [==============================] - 0s 4ms/step - loss:
32938630.0000
Epoch 391/500
2/2 [==============================] - 0s 6ms/step - loss:
34103348.0000
Epoch 392/500
2/2 [==============================] - 0s 6ms/step - loss:
33429920.0000
Epoch 393/500
2/2 [==============================] - 0s 5ms/step - loss:
32369866.0000
Epoch 394/500
2/2 [==============================] - 0s 5ms/step - loss:
32768036.0000
Epoch 395/500
2/2 [==============================] - 0s 6ms/step - loss:
34188352.0000
Epoch 396/500
2/2 [==============================] - 0s 6ms/step - loss:
34007336.0000
Epoch 397/500
2/2 [==============================] - 0s 5ms/step - loss:
32489274.0000
Epoch 398/500
2/2 [==============================] - 0s 5ms/step - loss:
31976950.0000
Epoch 399/500
2/2 [==============================] - 0s 7ms/step - loss:
32147322.0000
Epoch 400/500
2/2 [==============================] - 0s 6ms/step - loss:
32784864.0000
```

```
Epoch 401/500
2/2 [==============================] - 0s 4ms/step - loss:
33687088.0000
Epoch 402/500
2/2 [==============================] - 0s 5ms/step - loss:
32382150.0000
Epoch 403/500
2/2 [==============================] - 0s 7ms/step - loss:
31433814.0000
Epoch 404/500
2/2 [==============================] - 0s 6ms/step - loss:
33485900.0000
Epoch 405/500
2/2 [==============================] - 0s 4ms/step - loss:
35821644.0000
Epoch 406/500
2/2 [==============================] - 0s 6ms/step - loss:
34163996.0000
Epoch 407/500
2/2 [==============================] - 0s 6ms/step - loss:
31284668.0000
Epoch 408/500
2/2 [==============================] - 0s 5ms/step - loss:
32632806.0000
Epoch 409/500
2/2 [==============================] - 0s 5ms/step - loss:
34099576.0000
Epoch 410/500
2/2 [==============================] - 0s 7ms/step - loss:
33205926.0000
Epoch 411/500
2/2 [==============================] - 0s 6ms/step - loss:
31572598.0000
Epoch 412/500
2/2 [==============================] - 0s 5ms/step - loss:
32014698.0000
Epoch 413/500
2/2 [==============================] - 0s 6ms/step - loss:
32504908.0000
Epoch 414/500
2/2 [==============================] - 0s 6ms/step - loss:
31944874.0000
Epoch 415/500
2/2 [==============================] - 0s 5ms/step - loss:
31275782.0000
Epoch 416/500
2/2 [==============================] - 0s 5ms/step - loss:
31789376.0000
Epoch 417/500
2/2 [==============================] - 0s 7ms/step - loss:
```

```
30602096.0000
Epoch 418/500
2/2 [==============================] - 0s 6ms/step - loss:
31713184.0000
Epoch 419/500
2/2 [==============================] - 0s 5ms/step - loss:
33671176.0000
Epoch 420/500
2/2 [==============================] - 0s 6ms/step - loss:
33107706.0000
Epoch 421/500
2/2 [==============================] - 0s 7ms/step - loss:
30819158.0000
Epoch 422/500
2/2 [==============================] - 0s 5ms/step - loss:
30586570.0000
Epoch 423/500
2/2 [==============================] - 0s 5ms/step - loss:
31124588.0000
Epoch 424/500
2/2 [==============================] - 0s 7ms/step - loss:
30147844.0000
Epoch 425/500
2/2 [==============================] - 0s 6ms/step - loss:
29889732.0000
Epoch 426/500
2/2 [==============================] - 0s 4ms/step - loss:
33571480.0000
Epoch 427/500
2/2 [==============================] - 0s 6ms/step - loss:
31444516.0000
Epoch 428/500
2/2 [==============================] - 0s 7ms/step - loss:
30517988.0000
Epoch 429/500
2/2 [==============================] - 0s 4ms/step - loss:
31859562.0000
Epoch 430/500
2/2 [==============================] - 0s 5ms/step - loss:
31690224.0000
Epoch 431/500
2/2 [==============================] - 0s 6ms/step - loss:
29665148.0000
Epoch 432/500
2/2 [==============================] - 0s 4ms/step - loss:
29658934.0000
Epoch 433/500
2/2 [==============================] - 0s 5ms/step - loss:
29779802.0000
Epoch 434/500
```

```
2/2 [==============================] - 0s 6ms/step - loss:
29754314.0000
Epoch 435/500
2/2 [==============================] - 0s 5ms/step - loss:
29417856.0000
Epoch 436/500
2/2 [==============================] - 0s 5ms/step - loss:
30437236.0000
Epoch 437/500
2/2 [==============================] - 0s 5ms/step - loss:
30358490.0000
Epoch 438/500
2/2 [==============================] - 0s 5ms/step - loss:
28558300.0000
Epoch 439/500
2/2 [==============================] - 0s 5ms/step - loss:
29245530.0000
Epoch 440/500
2/2 [==============================] - 0s 6ms/step - loss:
32563412.0000
Epoch 441/500
2/2 [==============================] - 0s 6ms/step - loss:
31312500.0000
Epoch 442/500
2/2 [==============================] - 0s 5ms/step - loss:
28937140.0000
Epoch 443/500
2/2 [==============================] - 0s 5ms/step - loss:
31789708.0000
Epoch 444/500
2/2 [==============================] - 0s 6ms/step - loss:
36402456.0000
Epoch 445/500
2/2 [==============================] - 0s 5ms/step - loss:
34410264.0000
Epoch 446/500
2/2 [==============================] - 0s 5ms/step - loss:
30506074.0000
Epoch 447/500
2/2 [==============================] - 0s 6ms/step - loss:
31399542.0000
Epoch 448/500
2/2 [==============================] - 0s 6ms/step - loss:
31905754.0000
Epoch 449/500
2/2 [==============================] - 0s 5ms/step - loss:
29168694.0000
Epoch 450/500
2/2 [==============================] - 0s 5ms/step - loss:
30844000.0000
```

```
Epoch 451/500
2/2 [==============================] - 0s 6ms/step - loss:
31375584.0000
Epoch 452/500
2/2 [==============================] - 0s 6ms/step - loss:
28869648.0000
Epoch 453/500
2/2 [==============================] - 0s 5ms/step - loss:
30524044.0000
Epoch 454/500
2/2 [==============================] - 0s 6ms/step - loss:
34618040.0000
Epoch 455/500
2/2 [==============================] - 0s 6ms/step - loss:
30537766.0000
Epoch 456/500
2/2 [==============================] - 0s 5ms/step - loss:
29051866.0000
Epoch 457/500
2/2 [==============================] - 0s 5ms/step - loss:
35420616.0000
Epoch 458/500
2/2 [==============================] - 0s 6ms/step - loss:
32971770.0000
Epoch 459/500
2/2 [==============================] - 0s 4ms/step - loss:
29686164.0000
Epoch 460/500
2/2 [==============================] - 0s 5ms/step - loss:
33155540.0000
Epoch 461/500
2/2 [==============================] - 0s 6ms/step - loss:
32299552.0000
Epoch 462/500
2/2 [==============================] - 0s 6ms/step - loss:
28525300.0000
Epoch 463/500
2/2 [==============================] - 0s 4ms/step - loss:
30860880.0000
Epoch 464/500
2/2 [==============================] - 0s 5ms/step - loss:
34971032.0000
Epoch 465/500
2/2 [==============================] - 0s 7ms/step - loss:
32650388.0000
Epoch 466/500
2/2 [==============================] - 0s 6ms/step - loss:
28639108.0000
Epoch 467/500
2/2 [==============================] - 0s 5ms/step - loss:
```

```
30142778.0000
Epoch 468/500
2/2 [==============================] - 0s 6ms/step - loss:
31048938.0000
Epoch 469/500
2/2 [==============================] - 0s 5ms/step - loss:
29681068.0000
Epoch 470/500
2/2 [==============================] - 0s 5ms/step - loss:
29602522.0000
Epoch 471/500

2/2 [==============================] - 0s 5ms/step - loss:
28986698.0000
Epoch 472/500
2/2 [==============================] - 0s 6ms/step - loss:
28658938.0000
Epoch 473/500
2/2 [==============================] - 0s 6ms/step - loss:
28758948.0000
Epoch 474/500
2/2 [==============================] - 0s 4ms/step - loss:
29706368.0000
Epoch 475/500
2/2 [==============================] - 0s 6ms/step - loss:
29657092.0000
Epoch 476/500
2/2 [==============================] - 0s 5ms/step - loss:
28371226.0000
Epoch 477/500
2/2 [==============================] - 0s 5ms/step - loss:
29128202.0000
Epoch 478/500
2/2 [==============================] - 0s 4ms/step - loss:
29468544.0000
Epoch 479/500
2/2 [==============================] - 0s 7ms/step - loss:
28635014.0000
Epoch 480/500
2/2 [==============================] - 0s 5ms/step - loss:
28024636.0000
Epoch 481/500
2/2 [==============================] - 0s 5ms/step - loss:
28779030.0000
Epoch 482/500
2/2 [==============================] - 0s 6ms/step - loss:
29289590.0000
Epoch 483/500
2/2 [==============================] - 0s 6ms/step - loss:
28785818.0000
```

```
Epoch 484/500
2/2 [==============================] - 0s 5ms/step - loss:
27991142.0000
Epoch 485/500
2/2 [==============================] - 0s 6ms/step - loss:
28030228.0000
Epoch 486/500
2/2 [==============================] - 0s 7ms/step - loss:
28550298.0000
Epoch 487/500
2/2 [==============================] - 0s 10ms/step - loss:
28880048.0000
Epoch 488/500
2/2 [==============================] - 0s 7ms/step - loss:
29258196.0000
Epoch 489/500
2/2 [==============================] - 0s 6ms/step - loss:
29202604.0000
Epoch 490/500
2/2 [==============================] - 0s 6ms/step - loss:
28328118.0000
Epoch 491/500
2/2 [==============================] - 0s 5ms/step - loss:
28343084.0000
Epoch 492/500
2/2 [==============================] - 0s 6ms/step - loss:
27389914.0000
Epoch 493/500
2/2 [==============================] - 0s 8ms/step - loss:
27545386.0000
Epoch 494/500
2/2 [==============================] - 0s 7ms/step - loss:
28085494.0000
Epoch 495/500
2/2 [==============================] - 0s 5ms/step - loss:
28438276.0000
Epoch 496/500
2/2 [==============================] - 0s 6ms/step - loss:
27719268.0000
Epoch 497/500
2/2 [==============================] - 0s 7ms/step - loss:
27452474.0000
Epoch 498/500
2/2 [==============================] - 0s 5ms/step - loss:
27281204.0000
Epoch 499/500
2/2 [==============================] - 0s 5ms/step - loss:
27280700.0000
Epoch 500/500
```

```
2/2 [==============================] - 0s 6ms/step - loss:
26842260.0000

<keras.callbacks.History at 0x25830b31c60>
```

```
from numpy import array
# demonstrate prediction for next 10 days
x_input = array(df_17_cumm.cc.to_list()[-12:])
temp_input=list(x_input)
lst_output=[]
i=0
while(i<3):

    if(len(temp_input)>12):
        x_input=array(temp_input[1:])
#         print("{} day input {}".format(i,x_input))
        #print(x_input)
        x_input = x_input.reshape((1, n_steps, n_features))
        #print(x_input)
        yhat = model.predict(x_input, verbose=0)
#         print("{} day output {}".format(i,yhat))
        temp_input.append(yhat[0][0])
        temp_input=temp_input[1:]
        #print(temp_input)
        lst_output.append(yhat[0][0])
        i=i+1
    else:
        x_input = x_input.reshape((1, n_steps, n_features))
        yhat = model.predict(x_input, verbose=0)
#         print(yhat[0])
        temp_input.append(yhat[0][0])
        lst_output.append(yhat[0][0])
        i=i+1


print(lst_output)
```

```
[27170.455, 27841.54, 27784.979]
```

```
pred_data = lstm_data[-16:]
pred_data.reset_index(drop=True, inplace=True)
pred_data
```

	ds	y
0	1	44327.0
1	2	40076.0
2	3	38164.0
3	4	32852.0
4	5	27944.0
5	6	30685.0
6	7	30571.0

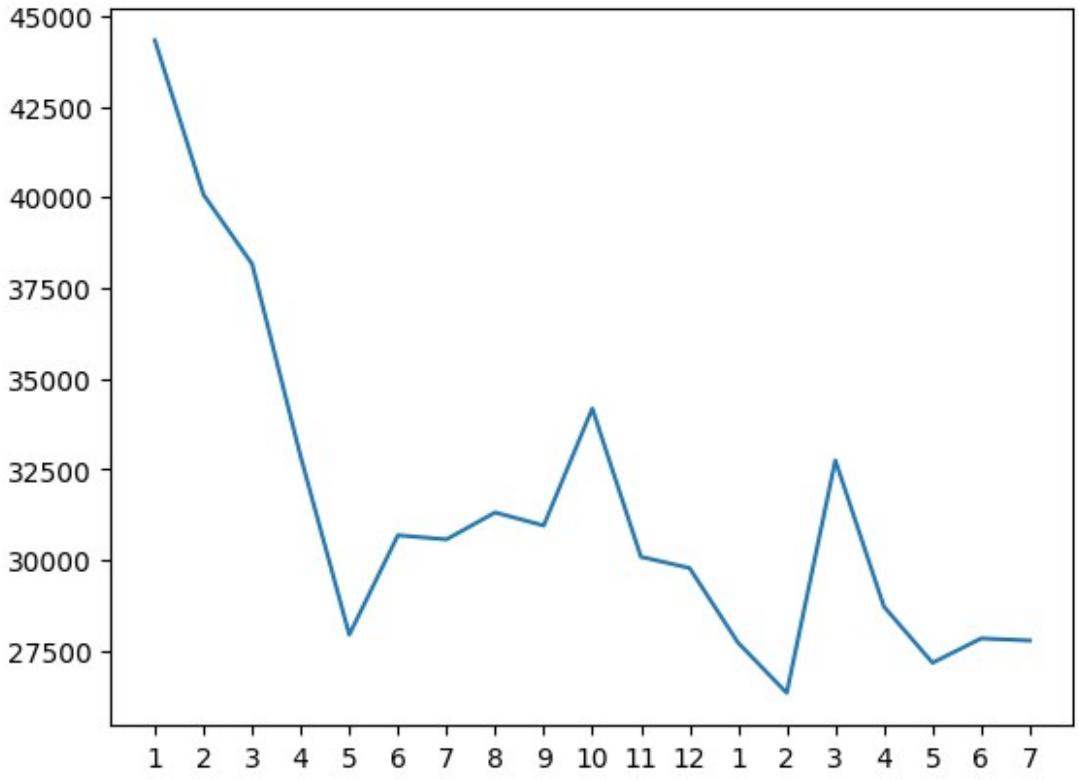
```
7    8  31314.0
8    9  30954.0
9   10  34179.0
10  11  30091.0
11  12  29783.0
12   1  27722.0
13   2  26341.0
14   3  32751.0
15   4  28718.0
```

```
tmp = pd.concat((pred_data, pd.DataFrame({
    "ds": [5, 6, 7],
    "y": lst_output
})))
```

```
tmp.reset_index(drop=True, inplace=True)
```

```
ax = plt.axes()
plt.plot(tmp.index, tmp.y)
ax.set_xticks(tmp.index, tmp.ds)
```

```
[<matplotlib.axis.XTick at 0x257eb7a9210>,
 <matplotlib.axis.XTick at 0x257eb7a9e40>,
 <matplotlib.axis.XTick at 0x257eb681060>,
 <matplotlib.axis.XTick at 0x258308dfd90>,
 <matplotlib.axis.XTick at 0x25834173f70>,
 <matplotlib.axis.XTick at 0x258341a0790>,
 <matplotlib.axis.XTick at 0x258341733d0>,
 <matplotlib.axis.XTick at 0x258341a0ee0>,
 <matplotlib.axis.XTick at 0x258341a16c0>,
 <matplotlib.axis.XTick at 0x258341a1ea0>,
 <matplotlib.axis.XTick at 0x258341a2680>,
 <matplotlib.axis.XTick at 0x25834172fe0>,
 <matplotlib.axis.XTick at 0x258341a1360>,
 <matplotlib.axis.XTick at 0x258341a30a0>,
 <matplotlib.axis.XTick at 0x258341a3880>,
 <matplotlib.axis.XTick at 0x258341a3160>,
 <matplotlib.axis.XTick at 0x258341c0880>,
 <matplotlib.axis.XTick at 0x258341a0cd0>,
 <matplotlib.axis.XTick at 0x258341c0a30>]
```




```
tmp[-3:]
```

```
    ds             y
16   5  27170.455078
17   6  27841.539062
18   7  27784.978516
```

```
pd.DataFrame({
    "Month":["01/05/2021", "01/06/2021", "01/07/2021"],
    "No_calls_Offered_predicted": tmp[-3:].y.to_list()
}).to_csv("month_pred.csv", index=False)
```

```
lstm_data
```

```
       ds        y
month
1       1  48239.0
```

2	2	42176.0
3	3	46218.0
4	4	42362.0
5	5	45914.0
6	6	45519.0
7	7	41117.0
8	8	42484.0
9	9	39754.0
10	10	46640.0
11	11	50725.0
12	12	44771.0
1	1	48219.0
2	2	44012.0
3	3	46840.0
4	4	48348.0
5	5	48600.0
6	6	43434.0
7	7	50356.0
8	8	54320.0
9	9	48260.0
10	10	58059.0
11	11	50977.0
12	12	44124.0
1	1	46429.0
2	2	43424.0
3	3	44893.0
4	4	48295.0
5	5	50901.0
6	6	46533.0
7	7	47944.0
8	8	47310.0
9	9	45262.0
10	10	49613.0
11	11	40431.0
12	12	37935.0
1	1	44327.0
2	2	40076.0
3	3	38164.0
4	4	32852.0
5	5	27944.0
6	6	30685.0
7	7	30571.0
8	8	31314.0
9	9	30954.0
10	10	34179.0
11	11	30091.0
12	12	29783.0
1	1	27722.0
2	2	26341.0

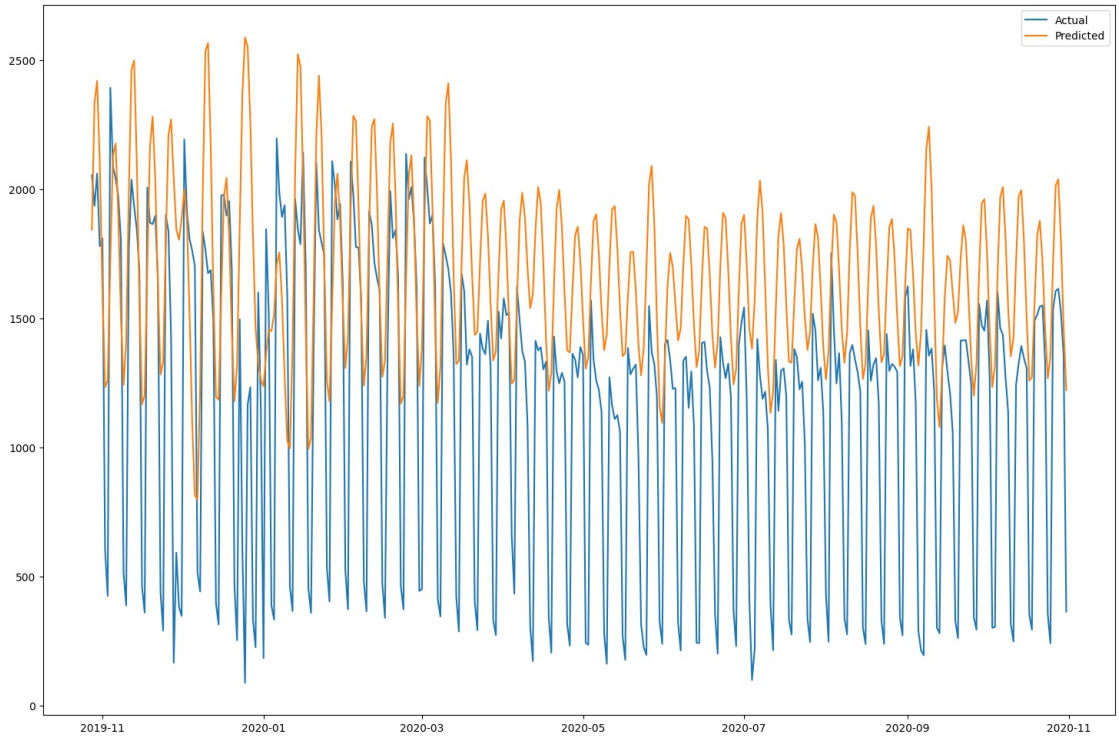
3	3	32751.0
4	4	28718.0

```
from sklearn.preprocessing import MinMaxScaler

scaler=MinMaxScaler(feature_range=(0,1))

all_data = lstm_data.values

all_data.shape

(1581, 2)

train_data = all_data[0:1000,:]
test_data = all_data[1000:1000+400,:]
val_data = all_data[1000+400:,:]

lstm_data.index = lstm_data.ds
lstm_data.drop('ds', axis = 1, inplace = True)

scaled_data = scaler.fit_transform(lstm_data)

x_train_data,y_train_data=[],[]

for i in range(30,len(train_data)):
    x_train_data.append(scaled_data[i-30:i,0])
    y_train_data.append(scaled_data[i,0])

x_train_data,y_train_data=np.array(x_train_data),np.array(y_train_data
)
x_train_data=np.reshape(x_train_data,
(x_train_data.shape[0],x_train_data.shape[1],1))

from keras.models import Sequential
from sklearn.pipeline import Pipeline
from keras.layers import LSTM, Dropout, Dense

lstm_model=Sequential()
lstm_model.add(LSTM(units=50,return_sequences=True,input_shape=(x_trai
n_data.shape[1],1)))
lstm_model.add(LSTM(units=50))
lstm_model.add(Dense(1))

inputs_data=lstm_data[len(lstm_data)-len(val_data)-
400:1000+400].values
inputs_data=inputs_data.reshape(-1,1)
inputs_data=scaler.transform(inputs_data)

lstm_model.compile(loss='mean_squared_error',optimizer='adam')
lstm_model.fit(x_train_data,y_train_data,epochs=1,batch_size=1,verbose
=1)
```

```
C:\Users\DV\AppData\Local\Programs\Python\Python310\lib\site-packages\
sklearn\base.py:450: UserWarning: X does not have valid feature names,
but MinMaxScaler was fitted with feature names
  warnings.warn(

970/970 [==============================] - 8s 6ms/step - loss: 0.0538

<keras.callbacks.History at 0x2579b389960>

X_test=[]
for i in range(30,inputs_data.shape[0]):
    X_test.append(inputs_data[i-30:i,0])
X_test=np.array(X_test)

X_test=np.reshape(X_test,(X_test.shape[0],X_test.shape[1],1))
predicted_closing_price=lstm_model.predict(X_test)
predicted_closing_price=scaler.inverse_transform(predicted_closing_pri
ce)

12/12 [==============================] - 0s 3ms/step

trd = lstm_data[0:1000]
tsd = lstm_data[1000+30:1000+400]

tsd['Pred'] = predicted_closing_price

C:\Users\DV\AppData\Local\Temp\ipykernel_19112\3216645329.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  tsd['Pred'] = predicted_closing_price

tsd

                 y         Pred
ds
2019-10-28  2053.0  1842.804199
2019-10-29  1935.0  2334.072021
2019-10-30  2060.0  2418.739746
2019-10-31  1778.0  2099.878418
2019-11-01  1809.0  1616.503540
...            ...          ...
2020-10-27  1605.0  2013.805908
2020-10-28  1614.0  2037.563477
2020-10-29  1522.0  1800.224731
2020-10-30  1355.0  1464.315430
2020-10-31   363.0  1221.064819
```

```
[370 rows x 2 columns]
```

```
fig, ax = plt.subplots(figsize=(18,12))
plt.plot(tsd.y, label='Actual')
plt.plot(tsd["Pred"], label='Predicted')
plt.legend()
plt.show()
```




```
predicted_closing_price.shape
```

```
(370, 1)
```

```
lstm_data[len(lstm_data)-len(val_data)-30:1000+400].values
```

```
array([[1269.],
       [ 300.],
       [ 303.],
       [1601.],
       [1461.],
       [1435.],
       [1273.],
       [1136.],
       [ 316.],
       [ 247.],
       [1243.],
       [1326.],
       [1392.],
       [1344.],
       [1307.],
       [ 351.],
       [ 293.],
       [1490.],
       [1513.],
       [1545.],
       [1549.],
       [1331.],
       [ 353.],
       [ 240.],
       [1534.],
       [1605.],
       [1614.],
       [1522.],
       [1355.],
       [ 363.]])
```

```
tsd
```

ds	y
2019-10-28	2053.0
2019-10-29	1935.0
2019-10-30	2060.0
2019-10-31	1778.0
2019-11-01	1809.0
...	...
2020-10-27	1605.0
2020-10-28	1614.0
2020-10-29	1522.0
2020-10-30	1355.0
2020-10-31	363.0

[370 rows x 1 columns]

```
predicted_closing_price
```

```
array([[1842.8042],
       [2334.072 ],
       [2418.7397],
       [2099.8784],
       [1616.5035],
       [1232.0956],
       [1257.5872],
       [1728.2781],
       [2129.961 ],
       [2176.6262],
       [1903.156 ],
       [1509.7295],
       [1241.9193],
       [1404.8439],
       [1979.5037],
       [2462.3196],
       [2497.6514],
       [2118.308 ],
       [1579.0979],
       [1166.1064],
       [1201.8636],
       [1698.9268],
       [2163.4016],
       [2280.781 ],
       [2044.0519],
       [1625.6841],
       [1281.4686],
       [1330.3325],
       [1794.6653],
       [2206.1448],
       [2269.3901],
       [2040.8314],
       [1843.9038],
       [1803.429 ],
       [1885.5533],
       [1998.3645],
       [1878.9131],
       [1517.6407],
       [1097.49  ],
       [ 812.5472],
       [ 798.4467],
       [1231.3523],
       [1988.3202],
       [2534.499 ],
       [2565.591 ],
       [2168.307 ],
       [1618.0852],
       [1194.744 ],
       [1183.6005],
       [1588.7065],
```

```
[1968.8473],
[2043.0049],
[1812.559 ],
[1445.0282],
[1177.5826],
[1314.5142],
[1861.7777],
[2365.741 ],
[2587.4539],
[2551.1477],
[2265.9905],
[1823.3485],
[1461.6445],
[1322.9962],
[1260.6814],
[1235.4147],
[1362.2063],
[1454.5012],
[1449.8452],
[1514.0104],
[1707.0868],
[1754.6198],
[1572.2903],
[1275.0234],
[1025.7407],
[ 995.7112],
[1380.5089],
[2066.6484],
[2522.1672],
[2475.3115],
[2035.5383],
[1442.2687],
[ 992.7454],
[1036.9396],
[1589.3208],
[2207.8   ],
[2439.056 ],
[2205.0757],
[1717.8741],
[1257.3066],
[1177.6782],
[1552.9911],
[1945.2794],
[2059.3516],
[1882.3632],
[1556.7161],
[1306.5004],
[1427.3046],
[1914.4003],
[2283.3386],
```

[2263.852],
[1936.4447],
[1516.9524],
[1237.7552],
[1355.4745],
[1844.9685],
[2242.2798],
[2271.0452],
[1979.8365],
[1575.3806],
[1272.779],
[1333.89],
[1760.6986],
[2178.4712],
[2253.6196],
[1974.9106],
[1531.8727],
[1168.14],
[1198.111],
[1649.6206],
[2053.499],
[2130.014],
[1882.1971],
[1500.763],
[1237.4937],
[1381.8905],
[1895.7028],
[2281.4932],
[2265.679],
[1922.6761],
[1473.0845],
[1171.8148],
[1312.6512],
[1863.0939],
[2331.9185],
[2409.041],
[2116.0852],
[1667.5338],
[1321.6646],
[1333.1586],
[1703.2738],
[2042.8416],
[2111.801],
[1939.8075],
[1666.0677],
[1434.757],
[1446.3113],
[1716.4467],
[1953.013],
[1982.4945],

[1819.3846],
[1556.1555],
[1336.0557],
[1370.15],
[1668.5532],
[1921.6023],
[1954.887],
[1767.937],
[1482.9679],
[1246.3861],
[1260.8279],
[1570.0367],
[1877.1528],
[1985.2535],
[1889.4132],
[1692.3191],
[1538.4303],
[1591.2692],
[1841.7073],
[2007.2377],
[1946.4246],
[1703.7854],
[1418.7712],
[1218.2283],
[1285.9474],
[1625.5076],
[1923.3431],
[1996.5365],
[1852.8864],
[1601.764],
[1374.7717],
[1367.4459],
[1603.1287],
[1819.1495],
[1854.5275],
[1719.0502],
[1498.6027],
[1304.6162],
[1348.3091],
[1641.2019],
[1876.402],
[1902.1451],
[1748.5623],
[1532.7448],
[1375.5444],
[1435.7578],
[1707.15],
[1920.7959],
[1933.7743],
[1771.5941],

[1539.695],
[1352.5608],
[1364.4028],
[1581.9276],
[1755.244],
[1756.6721],
[1605.0419],
[1397.4404],
[1278.1381],
[1385.7213],
[1687.064],
[2017.5553],
[2089.8542],
[1864.504],
[1489.7035],
[1157.133],
[1093.5781],
[1338.4525],
[1622.6432],
[1752.8752],
[1698.514],
[1550.1764],
[1413.7633],
[1461.0117],
[1706.2943],
[1896.082],
[1883.1294],
[1709.5692],
[1477.1929],
[1310.387],
[1372.5925],
[1644.981],
[1853.1542],
[1847.3806],
[1664.6636],
[1435.2206],
[1307.5685],
[1407.1649],
[1702.0333],
[1907.7428],
[1887.9559],
[1688.06],
[1427.4772],
[1243.5125],
[1303.2905],
[1605.9844],
[1865.7997],
[1899.972],
[1709.7798],
[1458.0253],

[1381.2747],
[1569.1487],
[1887.2993],
[2032.5555],
[1909.836],
[1618.5236],
[1313.1561],
[1132.6527],
[1212.4176],
[1535.5638],
[1818.6831],
[1906.5814],
[1784.6602],
[1547.024],
[1330.9447],
[1327.8611],
[1557.9438],
[1768.3351],
[1807.0327],
[1687.5721],
[1502.6737],
[1376.7645],
[1448.0198],
[1700.1794],
[1863.8723],
[1812.3942],
[1611.4869],
[1386.9567],
[1262.2961],
[1369.9116],
[1690.371],
[1900.6324],
[1871.631],
[1681.5397],
[1454.7981],
[1326.8036],
[1441.0348],
[1757.6669],
[1987.9034],
[1973.2537],
[1752.6255],
[1468.583],
[1264.3822],
[1321.4816],
[1629.9055],
[1887.2188],
[1935.2458],
[1781.0365],
[1530.7474],
[1329.2797],

[1359.9703],
[1628.2876],
[1853.1078],
[1883.8923],
[1731.8433],
[1502.2491],
[1314.0214],
[1352.4458],
[1627.8656],
[1847.4741],
[1840.935],
[1663.6067],
[1443.7241],
[1315.9263],
[1442.4025],
[1787.8264],
[2155.2234],
[2241.1553],
[1999.1086],
[1578.3341],
[1187.5983],
[1077.0708],
[1293.3052],
[1582.3252],
[1741.5997],
[1723.6248],
[1599.0316],
[1481.7395],
[1524.1106],
[1729.2114],
[1859.6667],
[1796.3907],
[1580.8563],
[1340.5491],
[1199.7336],
[1321.1046],
[1676.0248],
[1945.7473],
[1960.4651],
[1753.0511],
[1451.3439],
[1233.0548],
[1314.6271],
[1674.9874],
[1968.6493],
[2007.7185],
[1813.5199],
[1539.3978],
[1352.2886],
[1420.9695],

```
       [1721.5897],
       [1972.4407],
       [1995.8644],
       [1796.1997],
       [1500.8687],
       [1257.6969],
       [1271.1051],
       [1555.94  ],
       [1820.1274],
       [1876.9185],
       [1719.839 ],
       [1461.5863],
       [1266.836 ],
       [1351.2676],
       [1713.7836],
       [2013.8059],
       [2037.5635],
       [1800.2247],
       [1464.3154],
       [1221.0648]], dtype=float32)
```

```
lstm_data[1000+400:].shape
```

```
(181, 1)
```

```
(0,)
```

```
US_holidays_series = cal.holidays(start='2017-01-01', end='2021-07-
31', return_name=True)
US_holidays = US_holidays_series.to_frame("name")
US_holidays.reset_index(inplace=True)
US_holidays.rename(columns={"index":"date"}, inplace=True)
```

```
prophet_holiday_df = pd.DataFrame(columns=["holiday", "ds",
"lower_window", "upper_window"])
all_US_holidays = US_holidays.name.unique()
for h in all_US_holidays:
    print(h)
```

```
    if h == "Thanksgiving Day":
        df_hday = pd.DataFrame({
            'holiday': h,
            'ds': pd.to_datetime(US_holidays[US_holidays.name ==
h].date),
            'lower_window': 0,
            'upper_window': 1
        })
    elif h == "Christmas Day":
        df_hday = pd.DataFrame({
            'holiday': h,
            'ds': pd.to_datetime(US_holidays[US_holidays.name ==
h].date),
            'lower_window': -1,
            'upper_window': 0
        })
    else:
        df_hday = pd.DataFrame({
            'holiday': h,
            'ds': pd.to_datetime(US_holidays[US_holidays.name ==
h].date),
            'lower_window': 0,
            'upper_window': 0
        })
    df_hday.reset_index(drop=True, inplace=True)
    prophet_holiday_df = pd.concat((prophet_holiday_df, df_hday))
```

```
New Year's Day
Birthday of Martin Luther King, Jr.
Washington’s Birthday
Memorial Day
Independence Day
Labor Day
Columbus Day
Veterans Day
Thanksgiving Day
Christmas Day
Juneteenth National Independence Day
```

```
prophet_holiday_df
```

```
                              holiday         ds lower_window
upper_window
0                       New Year's Day 2017-01-02            0
0
1                       New Year's Day 2018-01-01            0
0
2                       New Year's Day 2019-01-01            0
0
3                       New Year's Day 2020-01-01            0
0
```

4	New Year's Day	2021-01-01	0
0			
0	Birthday of Martin Luther King, Jr.	2017-01-16	0
0			
1	Birthday of Martin Luther King, Jr.	2018-01-15	0
0			
2	Birthday of Martin Luther King, Jr.	2019-01-21	0
0			
3	Birthday of Martin Luther King, Jr.	2020-01-20	0
0			
4	Birthday of Martin Luther King, Jr.	2021-01-18	0
0			
0	Washington’s Birthday	2017-02-20	0
0			
1	Washington’s Birthday	2018-02-19	0
0			
2	Washington’s Birthday	2019-02-18	0
0			
3	Washington’s Birthday	2020-02-17	0
0			
4	Washington’s Birthday	2021-02-15	0
0			
0	Memorial Day	2017-05-29	0
0			
1	Memorial Day	2018-05-28	0
0			
2	Memorial Day	2019-05-27	0
0			
3	Memorial Day	2020-05-25	0
0			
4	Memorial Day	2021-05-31	0
0			
0	Independence Day	2017-07-04	0
0			
1	Independence Day	2018-07-04	0
0			
2	Independence Day	2019-07-04	0
0			
3	Independence Day	2020-07-03	0
0			
4	Independence Day	2021-07-05	0
0			
0	Labor Day	2017-09-04	0
0			
1	Labor Day	2018-09-03	0
0			
2	Labor Day	2019-09-02	0
0			
3	Labor Day	2020-09-07	0
0			

```
0                          Columbus Day 2017-10-09           0
0
1                          Columbus Day 2018-10-08           0
0
2                          Columbus Day 2019-10-14           0
0
3                          Columbus Day 2020-10-12           0
0
0                          Veterans Day 2017-11-10           0
0
1                          Veterans Day 2018-11-12           0
0
2                          Veterans Day 2019-11-11           0
0
3                          Veterans Day 2020-11-11           0
0
0                       Thanksgiving Day 2017-11-23           0
1
1                       Thanksgiving Day 2018-11-22           0
1
2                       Thanksgiving Day 2019-11-28           0
1
3                       Thanksgiving Day 2020-11-26           0
1
0                          Christmas Day 2017-12-25          -1
0
1                          Christmas Day 2018-12-25          -1
0
2                          Christmas Day 2019-12-25          -1
0
3                          Christmas Day 2020-12-25          -1
0
0  Juneteenth National Independence Day 2021-06-18           0
0
```

```
all_Manila_holidays = manila_cal.holidays(start="2017-01-
01",end="2021-07-31",return_name=True)
all_m_holidays = all_Manila_holidays.to_frame("name")
all_m_holidays.reset_index(inplace=True)
all_m_holidays.rename(columns={"index":"date"}, inplace=True)

all_Man_holidays = all_m_holidays.name.unique()
for h in all_Man_holidays:
    print(h)
    df_hday = pd.DataFrame({
        'holiday': "Manila " + str(h),
        'ds': pd.to_datetime(holidays[holidays.name == h].date),
        'lower_window': 0,
        'upper_window': 0
    })
```

```
    df_hday.reset_index(drop=True, inplace=True)
    prophet_holiday_df = pd.concat((prophet_holiday_df, df_hday))
```

```
New Year
Day of Valor
Maundy Thursday
Good Friday
Labor Day
Independence Day
National heroes day
Bonifacio
Christmas
Rizal
```

```
prophet_holiday_df
```

```
                                holiday         ds lower_window
upper_window
0                        New Year's Day 2017-01-02            0
0
1                        New Year's Day 2018-01-01            0
0
2                        New Year's Day 2019-01-01            0
0
3                        New Year's Day 2020-01-01            0
0
4                        New Year's Day 2021-01-01            0
0
0   Birthday of Martin Luther King, Jr. 2017-01-16            0
0
1   Birthday of Martin Luther King, Jr. 2018-01-15            0
0
2   Birthday of Martin Luther King, Jr. 2019-01-21            0
0
3   Birthday of Martin Luther King, Jr. 2020-01-20            0
0
4   Birthday of Martin Luther King, Jr. 2021-01-18            0
0
0                Washington’s Birthday 2017-02-20            0
0
1                Washington’s Birthday 2018-02-19            0
0
2                Washington’s Birthday 2019-02-18            0
0
3                Washington’s Birthday 2020-02-17            0
0
4                Washington’s Birthday 2021-02-15            0
0
0                         Memorial Day 2017-05-29            0
0
1                         Memorial Day 2018-05-28            0
```

2	Memorial Day	2019-05-27	0
3	Memorial Day	2020-05-25	0
4	Memorial Day	2021-05-31	0
0	Independence Day	2017-07-04	0
1	Independence Day	2018-07-04	0
2	Independence Day	2019-07-04	0
3	Independence Day	2020-07-03	0
4	Independence Day	2021-07-05	0
0	Labor Day	2017-09-04	0
1	Labor Day	2018-09-03	0
2	Labor Day	2019-09-02	0
3	Labor Day	2020-09-07	0
0	Columbus Day	2017-10-09	0
1	Columbus Day	2018-10-08	0
2	Columbus Day	2019-10-14	0
3	Columbus Day	2020-10-12	0
0	Veterans Day	2017-11-10	0
1	Veterans Day	2018-11-12	0
2	Veterans Day	2019-11-11	0
3	Veterans Day	2020-11-11	0
0	Thanksgiving Day	2017-11-23	0
1	Thanksgiving Day	2018-11-22	0
2	Thanksgiving Day	2019-11-28	0
3	Thanksgiving Day	2020-11-26	0
0	Christmas Day	2017-12-25	-1

```
0
1                         Christmas Day 2018-12-25           -1
0
2                         Christmas Day 2019-12-25           -1
0
3                         Christmas Day 2020-12-25           -1
0
0  Juneteenth National Independence Day 2021-06-18            0
0
0                       Manila Labor Day 2017-09-04            0
0
1                       Manila Labor Day 2018-09-03            0
0
2                       Manila Labor Day 2019-09-02            0
0
3                       Manila Labor Day 2020-09-07            0
0
0                 Manila Independence Day 2017-07-04           0
0
1                 Manila Independence Day 2018-07-04           0
0
2                 Manila Independence Day 2019-07-04           0
0
3                 Manila Independence Day 2020-07-03           0
0
```

```python
dates17 = df_17.date.values
dates_not17 = df_not17_sorted.date.values
all_dates = list(dates17)
all_dates.extend(list(dates_not17))
# print(all_dates)
all_dates = pd.to_datetime(all_dates, format="%d/%m/%Y")
# print(all_dates)

cc17 = df_17.cc.values
cc_not17 = df_not17_sorted.cc.values
all_cc = list(cc17)
all_cc.extend(cc_not17)
# print(all_cc)

fbp_data = pd.DataFrame({
    "ds": all_dates,
    "y": all_cc
})
fbp_data
```

```
           ds       y
0    2017-01-01   264.0
1    2017-01-02   528.0
2    2017-01-03  2505.0
3    2017-01-04  2145.0
```

```
4    2017-01-05  2164.0
...         ...     ...
1576 2021-04-26  1392.0
1577 2021-04-27  1336.0
1578 2021-04-28  1220.0
1579 2021-04-29  1094.0
1580 2021-04-30  1119.0

[1581 rows x 2 columns]
```

```
fbp_model = Prophet(holidays=prophet_holiday_df)
```

```
fbp_model_fit = fbp_model.fit(fbp_data)
```

```
03:00:39 - cmdstanpy - INFO - Chain [1] start processing
03:00:39 - cmdstanpy - INFO - Chain [1] done processing
```

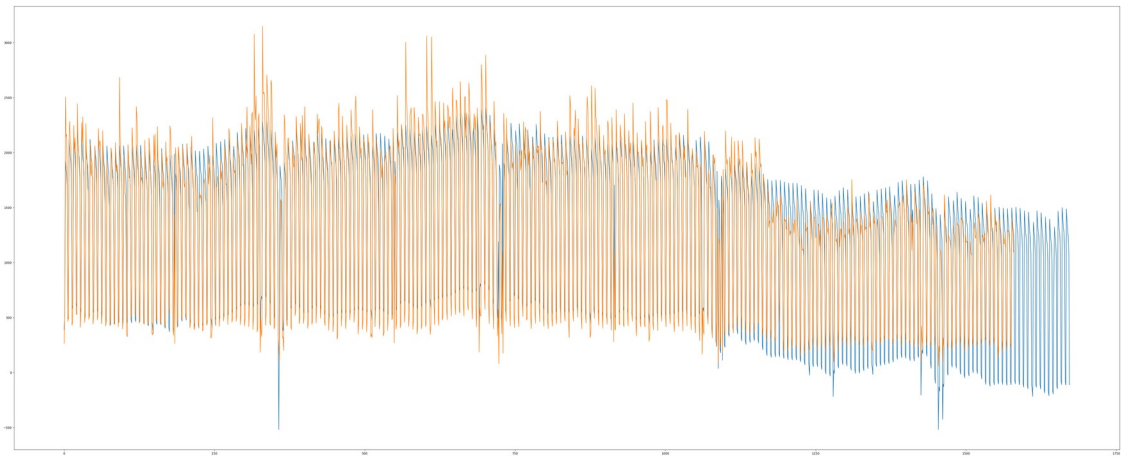
```
future = fbp_model_fit.make_future_dataframe(periods=92)
```

```
forecast = fbp_model_fit.predict(future)
```

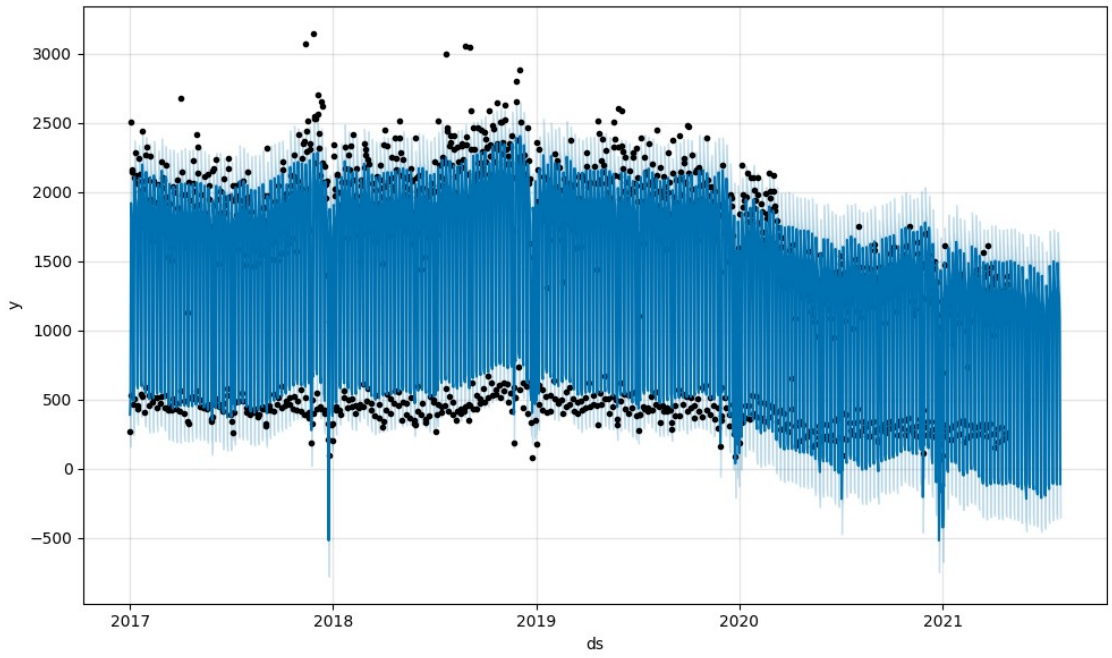
```
forecast[['ds', 'yhat']]
```

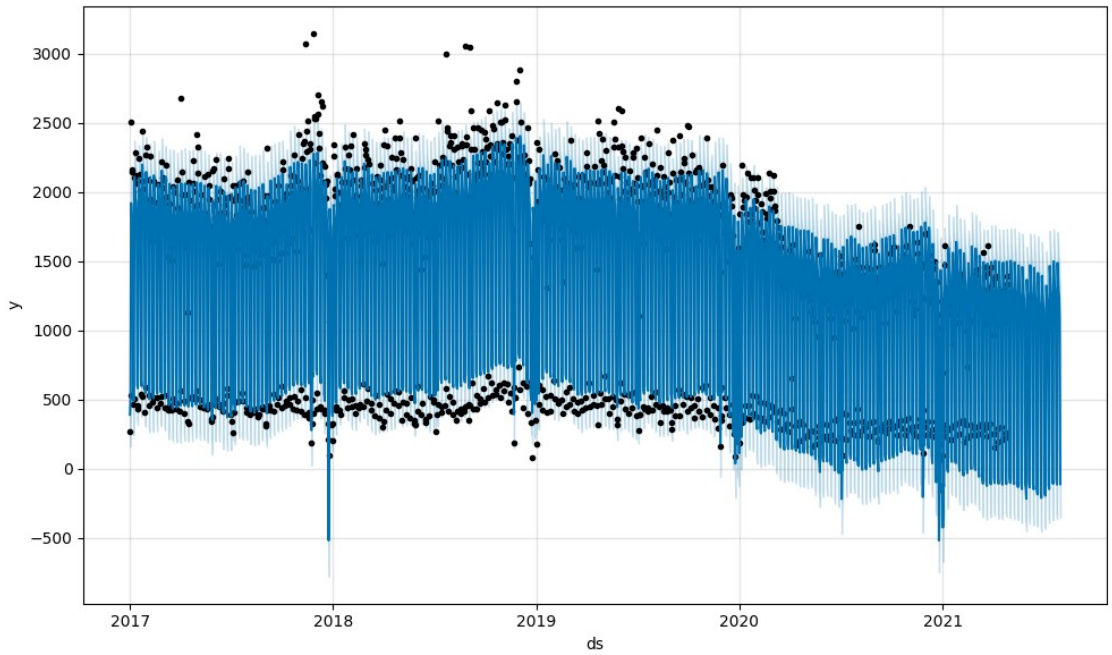
```
             ds         yhat
0    2017-01-01   390.268378
1    2017-01-02   510.781987
2    2017-01-03  1919.418719
3    2017-01-04  1844.600461
4    2017-01-05  1818.181766
...         ...          ...
1668 2021-07-27  1382.759824
1669 2021-07-28  1289.212060
1670 2021-07-29  1242.581627
1671 2021-07-30  1081.108678
1672 2021-07-31  -111.662933

[1673 rows x 2 columns]
```

```
fig = plt.figure()
fig.set_figheight(20)
fig.set_figwidth(50)
ax = fig.add_axes([1,1,1,1])
x = range(len(forecast))
x_org = range(len(fbp_data))
y_fbp = list(forecast.yhat.values)
y_org = all_cc
ax.plot(x, y_fbp)
ax.plot(x_org, all_cc)
plt.show()
```



```
fbp_model_fit.plot(forecast)
```






```
# [ arrow.get(dt, "DD/MM/YYYY").day for dt in df_17['date'] ]


# df['Date'] = df['Date'].apply(lambda row: arrow.get(str(row),
"M/D/YYYY").format("DD/MM/YYYY"))
```

```
# df.info()

# df['Year'] = df['Date'].apply(lambda row: arrow.get(row,
"DD/MM/YYYY").year)

# df_17 = df[df['Year'] == 2017]

# df_17

# df_17.rename(columns = {"Date":"date",
"No_of_calls_from_commercial_card":"cc",
"No_of_calls_from_retail_card":"rc",
"No_of_calls_from_student_card":"sc", "Year":"year"}, inplace=True)

# df_17

# plt.figure(figsize=(20, 20))
# plt.plot(df_17['date'], df_17['cc'], c='red', label='cc')
# plt.plot(df_17['date'], df_17['rc'], c='blue', label='rc')
# plt.plot(df_17['date'], df_17['sc'], c='green', label='sc')
# plt.legend()
# plt.show()

# df_17[['cc', 'rc', 'sc']]

# # (df_17.cc < df_17.rc).sum()
# ic(df_17.cc / df_17.rc, df_17.rc / df_17.sc, df_17.cc / df_17.rc ==
df_17.rc / df_17.sc)

# df_not17 = df[df['Year'] != 2017]

# df_not17

# df_not17.rename(columns = {"Date":"date",
"No_of_calls_from_commercial_card":"cc",
"No_of_calls_from_retail_card":"rc",
"No_of_calls_from_student_card":"sc", "Year":"year"}, inplace=True)

# df_not17

# plt.figure(figsize=(20, 20))
# plt.plot(df_not17['date'], df_not17['cc'], c='red', label='cc')
# plt.plot(df_not17['date'], df_not17['rc'], c='blue', label='rc')
# plt.plot(df_not17['date'], df_not17['sc'], c='green', label='sc')
# plt.legend()
# plt.show()

# df_18 = df[df['Year'] == 2018]
# df_18.rename(columns = {"Date":"date",
```

```
"No_of_calls_from_commercial_card":"cc",
"No_of_calls_from_retail_card":"rc",
"No_of_calls_from_student_card":"sc", "Year":"year"}, inplace=True)
# plt.figure(figsize=(20, 20))
# plt.plot(df_18['date'], df_18['cc'], c='red', label='cc')
# plt.plot(df_18['date'], df_18['rc'], c='blue', label='rc')
# plt.plot(df_18['date'], df_18['sc'], c='green', label='sc')
# plt.legend()
# plt.show()

# df_18

# df_not17

# df_not17.reset_index(inplace=True)

# df_not17.drop("index", axis=1, inplace=True)

# df_not17

# df_17.to_csv("df_17.csv")

# df_not17_sorted = pd.DataFrame(columns=['date', 'cc', 'rc', 'sc'])

# for index, row in df_not17.iterrows():
#     values = sorted([row.cc, row.rc, row.sc])
#     row_data = [row.date]
#     row_data.extend(values)
# #     print(row_data)
#     df_not17_sorted.loc[index] = row_data

# df_not17_sorted

# plt.figure(figsize=(30, 30))
# plt.plot(df_not17_sorted['date'], df_not17_sorted['cc'], c='red',
label='cc')
# plt.plot(df_not17_sorted['date'], df_not17_sorted['rc'], c='blue',
label='rc')
# plt.plot(df_not17_sorted['date'], df_not17_sorted['sc'], c='green',
label='sc')
# plt.legend()
# plt.show()

# ic(df_17.cc / df_17.rc, df_17.rc / df_17.sc, df_17.cc / df_17.rc ==
df_17.rc / df_17.sc)
# ic(df_not17_sorted.cc / df_not17_sorted.rc, df_not17_sorted.rc /
df_not17_sorted.sc, df_not17_sorted.cc / df_not17_sorted.rc ==
df_not17_sorted.rc / df_not17_sorted.sc)
```

```
# from math import round

# round(df_17.cc / df_17.rc, 2) == round(df_not17_sorted.cc[:365] /
df_not17_sorted.rc[:365], 2)

# for a, b in zip(df_17.cc / df_17.rc, df_not17_sorted.cc[:365] /
df_not17_sorted.rc[:365]):
#     print('%.1f'%a == '%.1f'%b)

# df_not17_sorted.to_csv("df_not17_sorted.csv")
```