Rising Concerns of Robocalls in the U.S.

Capstone Project #1

Sangyeol Baek

Problem Statement

In America, telephones have become a necessity in our lives. Not too recently, telephones have gone mobile so that we can carry them around with us on the go so that we do not miss a call from people we know from our colleagues or family members. Need to make a last-minute update to your significant other? No problem. Just call him or her right away. Stuck at home due to a pandemic from all of your friends and/or relatives? No problem. Our phones are here to keep us in touch. Need a day off due to sudden illness? No problem. Just contact your employer. Because of this, long-distance communication is basically a non-issue in America.

However, since we can call practically anyone, chances are you have encountered a call from a phone number or someone you do not recognize. Rarely, even the caller ID may be unknown. Suspiciously, when you pick up, most often you hear a peculiar robotic voice with various purposes ranging from the non-malicious telemarketing to the malicious scams that can drive unsuspecting people to exposing their personal or financial information. In much rarer but more severe cases, the scammer you might encounter is an actual person from another country. The federal government actively combats against such calls (loosely) called “robocalls.” Thus, by finding trends of robocalls in America this project hopes to make combating them easier.

Proposal

Potential Clients

- Phone users - the most obvious beneficiaries, who would use this analysis to better identify an unknown number
- FCC - the governmental group who actively combats robocalls and other unwanted phone calls would benefit from this analysis so that it would help the government battle robocalls more effectively

Data

The data provided is from the Kaggle site, which provided a link to the original source, the FCC. Here, the table contains more than 1.7M rows of data, each containing date of issue, phone number, location, with dates beginning mid-2014.

Approach

As I progress through the course, I should have more and more refined descriptions of specific models. However, on a higher level, some things I would like to address:

- How many robocalls can we expect in <insert number> years?

- What trends can we see regarding robocalls?
  - What kind of robocall?
  - Location and/or area code?
- Which kind of robocalls should we watch out for in the future?
- Is the number of robocalls increasing as time goes on?

Deliverables

The project will be available through the Github repository, which will contain a Jupyter notebook containing an engaging data storytelling with the relevant data visualizations and analyses, and a slideshow containing a general summary of the project.

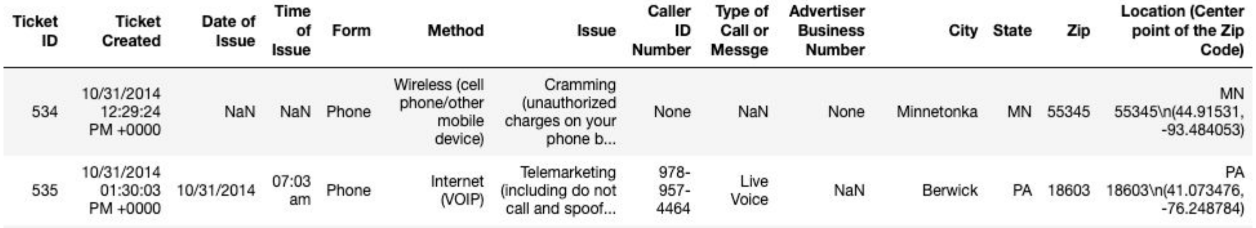
The Data

Data Collection

The data I obtained was from FCC and Kaggle, where the latter had only part of the data from the FCC database. To decide which one to use, I explored through each dataset in a temporary Jupyter notebook. Surprisingly, both datasets have approximately the same proportions of missing values. Ultimately, I chose the FCC's version since it's constantly updated and has a wider date range than Kaggle's.

Data Wrangling

Since the dataset largely consisted of text-based data, wrangling this data became a tricky task.



Ticket ID	Ticket Created	Date of Issue	Time of Issue	Form	Method	Issue	Caller ID Number	Type of Call or Messge	Advertiser Business Number	City	State	Zip	Location (Center point of the Zip Code)
534	10/31/2014 12:29:24 PM +0000	NaN	NaN	Phone	Wireless (cell phone/other mobile device)	Cramming (unauthorized charges on your phone b...	None	NaN	None	Minnetonka	MN	55345	MN 55345\n(44.91531, -93.484053)
535	10/31/2014 01:30:03 PM +0000	10/31/2014	07:03 am	Phone	Internet (VOIP)	Telemarketing (including do not call and spoof...	978-957-4464	Live Voice	NaN	Berwick	PA	18603	PA 18603\n(41.073476, -76.248784)

The most relevant columns of any numerical sort were the timestamp columns:

- Ticket Created
- Date of Issue
- Time of Issue

While "**Zip**" had numerical data, I decided to leave it out since it largely corresponds to the "**City**" and "**State**" columns.

As depicted above, while the “Ticket Created” column has NO missing values, the other two do, and additionally the dates and/or times could be different from that from the “Ticket Created” column. To combine the three columns, I created a new column called “Issue Occurred,” which will contain the date and time an issue occurred. For the missing dates and/or times, I used the dates and/or times from the “Ticket Created” columns to fill them out. For example:

Ticket Created	**Date of Issue**	**Time of Issue**
10/31/2014 12:29:24 PM +0000	NaN	NaN

=> **Issue Occurred**: 2014-10-31 12:29:24 PM

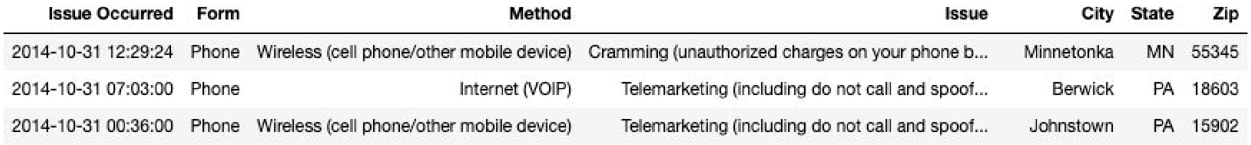
10/31/2014 01:30:03 PM +0000	10/31/2014	07:03 am

=> **Issue Occurred**: 2014-10-31 07:03:00 AM

After that, since the focus of this project is on robocalls, I selected out the rows in which the entries of the column “Form” matched “Phone.”

```
phone_df[phone_df.Form == 'Phone']
```

Next, I filtered out irrelevant columns and filtered out rows in which all three of “City”, “State” and “Zip” columns had empty values. Finally, I filled the remaining values with “Unknown.” The resulting dataset looked something like this:



Issue Occurred	Form	Method	Issue	City	State	Zip
2014-10-31 12:29:24	Phone	Wireless (cell phone/other mobile device)	Cramming (unauthorized charges on your phone b...	Minnetonka	MN	55345
2014-10-31 07:03:00	Phone	Internet (VOIP)	Telemarketing (including do not call and spoof...	Berwick	PA	18603
2014-10-31 00:36:00	Phone	Wireless (cell phone/other mobile device)	Telemarketing (including do not call and spoof...	Johnstown	PA	15902

Challenges

Needless to say, data will start as a cluttered mess. Ideally, it would be clean enough to grant us a head start. However, very often this is not practical. The dataset for this project is no exception. In fact, one of the biggest challenges was finding a way to combine the three time-based columns into one new column. Luckily, the columns “Ticket Created” and “Date of Issue” all had entries with a consistent format; “Time of Issue” did not. Looking closer to the latter column, the entries came in various formats:
(these are arbitrarily chosen to demonstrate format inconsistencies)

```
07:03 am, 12:36 PM, 12:30 A.M., 4:13 P.M., 05:02pm, 9:6 P.M., etc.
```

Because of this, trying to make them into datetime objects was cumbersome. Luckily, all had the colon separator and the AM/PM labels. To tackle this, I removed all whitespaces and formatted the AM/PM parts to have consistent formats:

```
07:03AM, 12:36PM, 12:30AM, 4:13PM, 05:02PM, 9:6PM, etc.
```

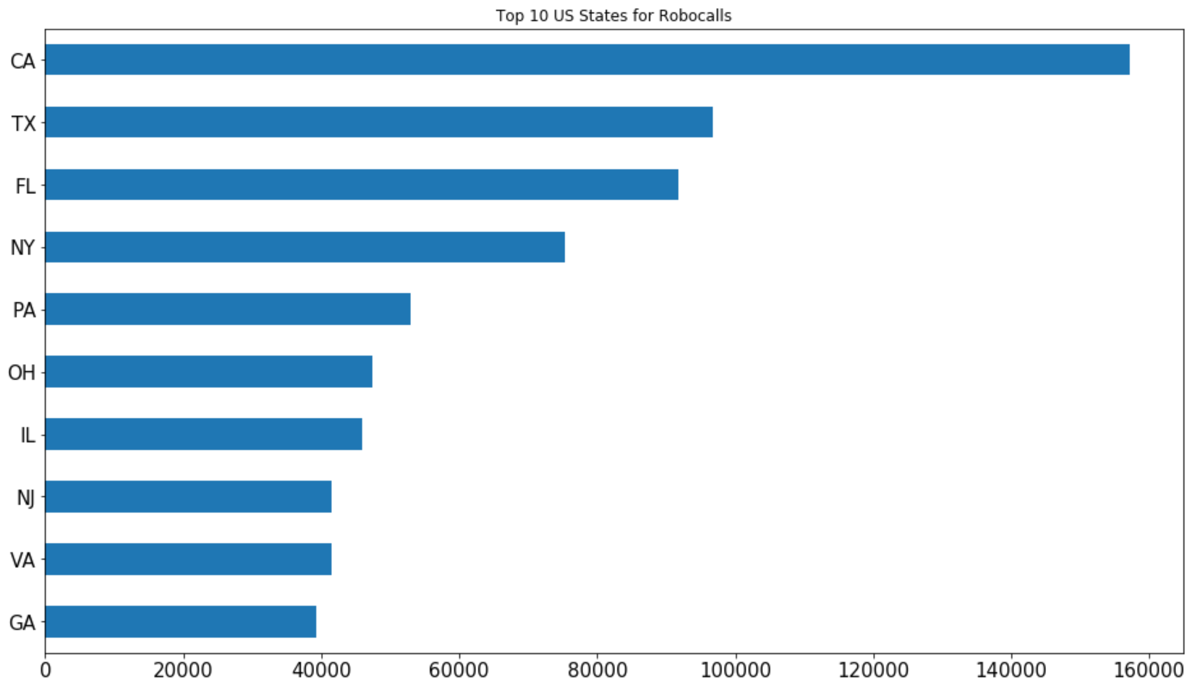
After that, I am able to create a column of times with consistent format.

However, another challenge I encountered came from trying to fill out the missing values for the date and time of issue columns. The idea was to fill them with the time and/or date from the “Ticket Created” column; however, I ran into trouble trying to use the “fillna” function provided by the Pandas library as all of the date and time columns were filled with the values from the latter regardless of whether there were missing values. Eventually, I decided to go safe and create three numpy arrays from each of the columns and manually fill in the values for each missing value. After that, I would create a new column array that would combine the date and time columns into one datetime column, “Issue Occurred.”

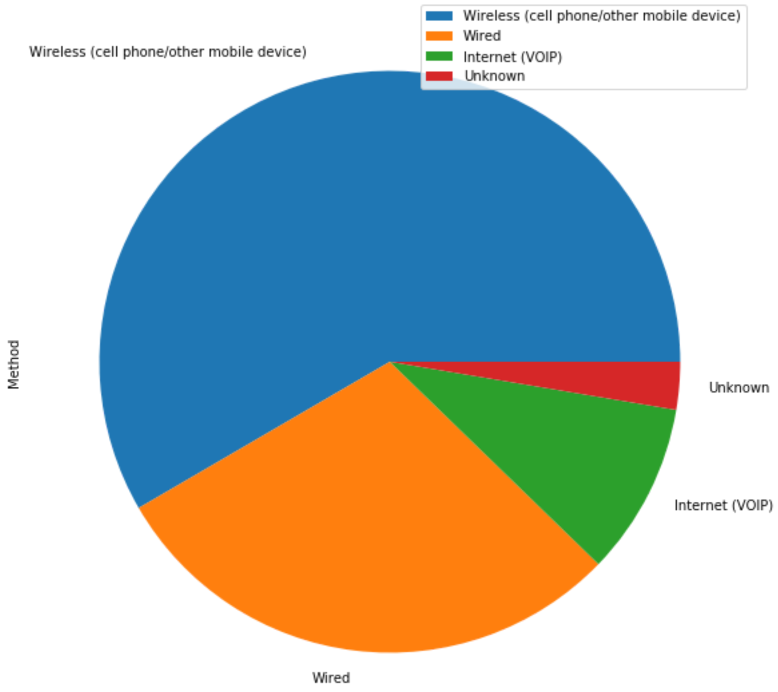
The challenges I encountered after data wrangling was finding use of the newly cleaned dataset. As aforementioned, the dataset I used hardly had any numerical values in the data, which limited my options especially from those designed to test correlations of two different variables. Luckily, I was able to use the timestamps in the dataset as populations, although it also meant I had to wrangle the data a bit more to obtain more numerical values.

Visualizing the Data

Below is a brief summary of the findings from the wrangled dataset.



Unsurprisingly, the robocalls tend to target the regions with greater population, which are also subjectively the most well known regions in America. Therefore, there is no doubt that California becomes the most prominent target (by a long run), with Texas at a distant second.



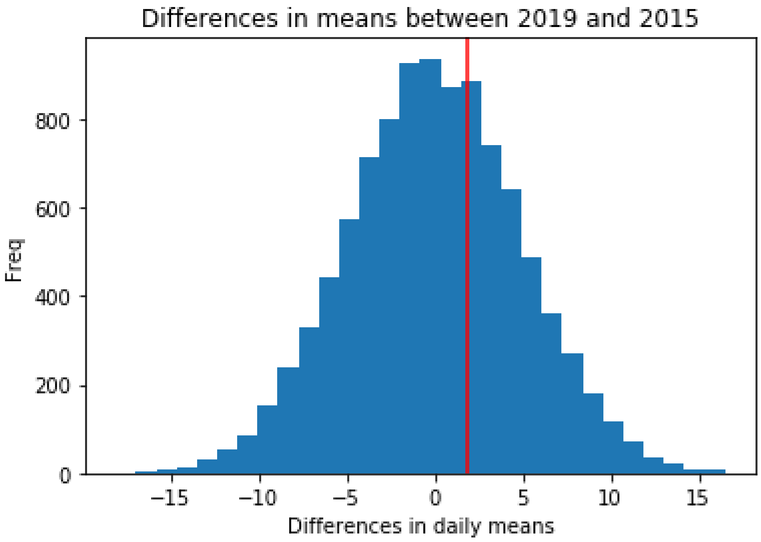
Since the number of wireless users, especially mobile phone users, increases overtime, it is also no surprise that most robocalls target wireless users.

Statistical Inferences

One of the questions I asked myself was whether the number of robocalls is increasing overtime? In other words, is there a statistically significant increase in the daily number of robocalls between 2015 and 2019?

$$H_0 : \mu_{2019} - \mu_{2015} \leq 0$$
$$H_a : \mu_{2019} - \mu_{2015} > 0$$

Using the bootstrap method, I tested the difference of means with the years as populations with the t-test with $\alpha = 0.05$ as the **one-tailed** statistical significance.



The red line indicates the observed difference of means under the null hypothesis H_0. This test resulted in $p = 0.352 > \alpha$, meaning we cannot reject the null hypothesis.

Another question I asked is if there is an increase in proportions for complaints from **wireless users** in 2019 from 2015. Once again, I used the data from 2015 and 2019 as the populations to test.

$$H_0 : p_{2019} - p_{2015} > 0$$
$$H_0 : p_{2019} - p_{2015} \leq 0$$

To test the difference of proportions, since each population is large enough, I can use the z-test with $\alpha = 0.05$ as the **one-tailed** statistical significance. We compute the values as followed:

$$\widehat{p}_{2015} = \frac{\text{\# Wireless in 2015}}{\text{\# Entries in 2015}} = \frac{k_{2015}}{n_{2015}} = 0.5163$$
$$\widehat{p}_{2019} = \frac{\text{\# Wireless in 2019}}{\text{\# Entries in 2019}} = \frac{k_{2019}}{n_{2019}} = 0.5027$$
$$\widehat{p} = \frac{k_{2015} + k_{2019}}{n_{2015} + n_{2019}} = 0.5095$$

Then, I compute the z-test:

$$z = \frac{\hat{p}_{2019} - \hat{p}_{2015}}{\sqrt{\hat{p}(1 - \hat{p})(\frac{1}{n_{2015}} + \frac{1}{n_{2019}})}}$$

Since $\alpha = 0.05$, the critical value must be $z^* = 1.64$. In order to reject the null hypothesis, we need to have $z > z^*$. However, I got $z = -1.92$, which we clearly cannot reject the null hypothesis. In fact, surprisingly, there seemed to be less wireless complaints proportionally in 2019 than in 2015.

While there are certainly other ways to use this dataset, the limited numerical data it provided rendered this impractical.

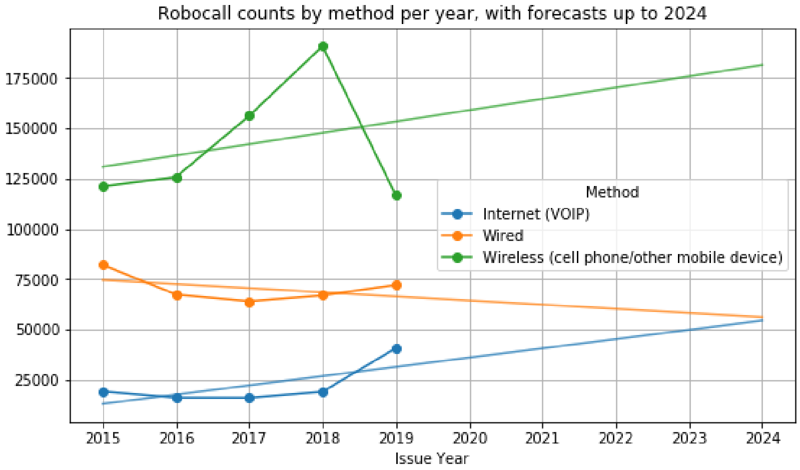
Deeper Analysis with Machine Learning

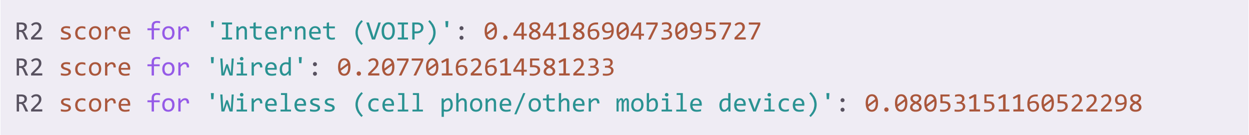
For this project, I would use Machine Learning to answer the question whether robocalls will rise or decline in the next few years. More specifically, which methods (namely the Wired, Internet (VOIP), and Wireless) used for robocalls will become the next trend and which ones will start declining? The first method I used was Linear Regression from the scikit-learn library.

Linear Regression

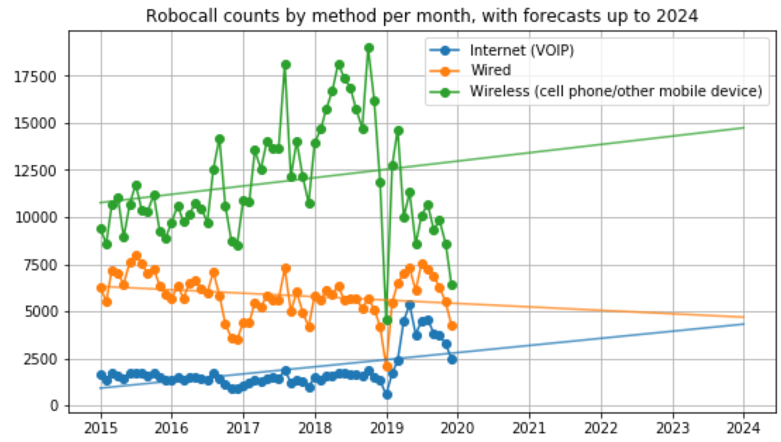
For this, I wrangled the dataset to aggregate by each method, with the date being the index. I created a Linear model for each different method grouped by YEARLY, MONTHLY, and WEEKLY totals and compared their R^2 scores.

The plots and results can be found on the next page.

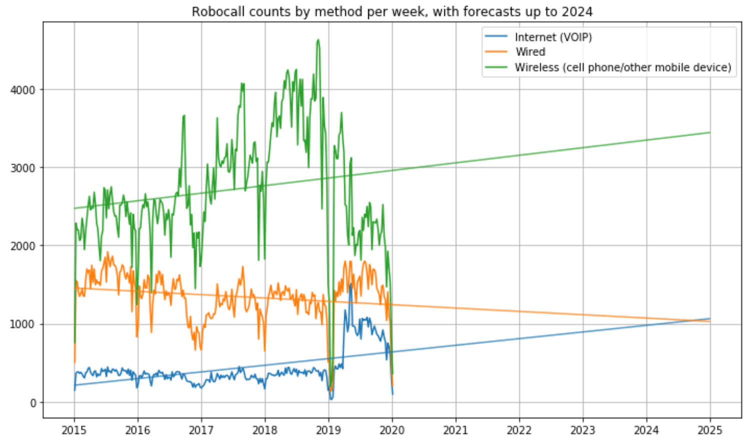




```
R2 score for 'Internet (VOIP)': 0.48418690473095727
R2 score for 'Wired': 0.20770162614581233
R2 score for 'Wireless (cell phone/other mobile device)': 0.08053151160522298
```



```
R2 score for 'Internet (VOIP)': 0.30160794562399384
R2 score for 'Wired': 0.054072006931844885
R2 score for 'Wireless (cell phone/other mobile device)': 0.04569825951258666
```



```
R2 score for 'Internet (VOIP)': 0.2721124359043263
R2 score for 'Wired': 0.04184601590632475
R2 score for 'Wireless (cell phone/other mobile device)': 0.0334525969207593
```

From the plots, even with the smaller time intervals, the linear models all produce similar results. Additionally, the yearly model performed best according to the R^2 scores for each robocall method.

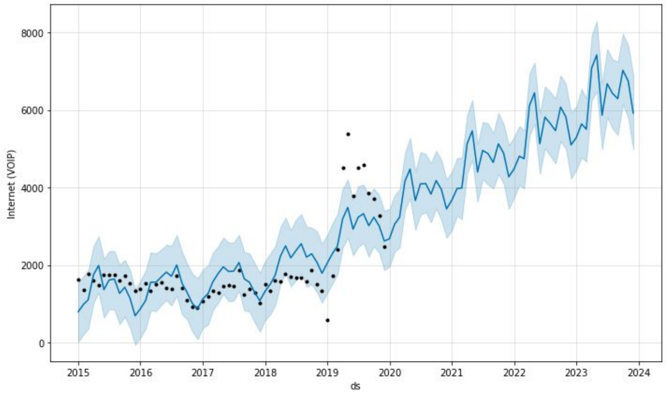
I figured linear regression is not so useful for time-based datasets after all. Fortunately, there is another library designed for time-based datasets built by Facebook, FBProphet.

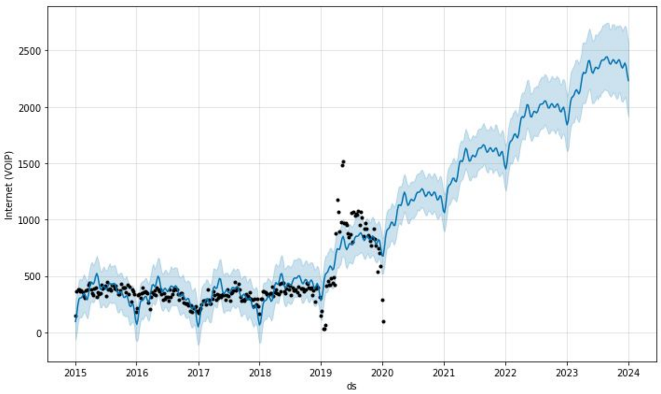
FBProphet

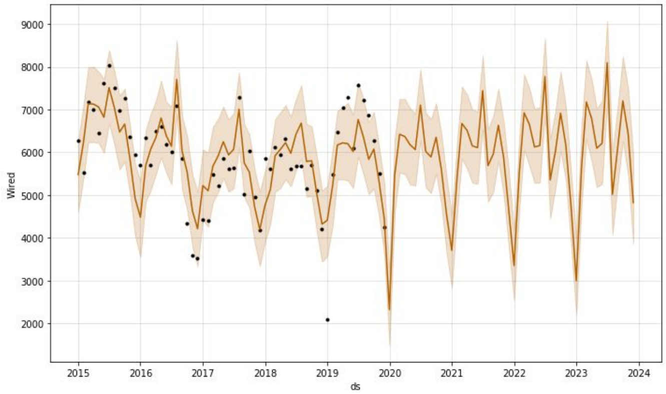
The second method FBProphet is an API that creates a forecasting dataframe based on the historical data provided by a time-based dataframe. It can be used to create more robust predictions than a linear model can, and it can be plotted for better visualizations.

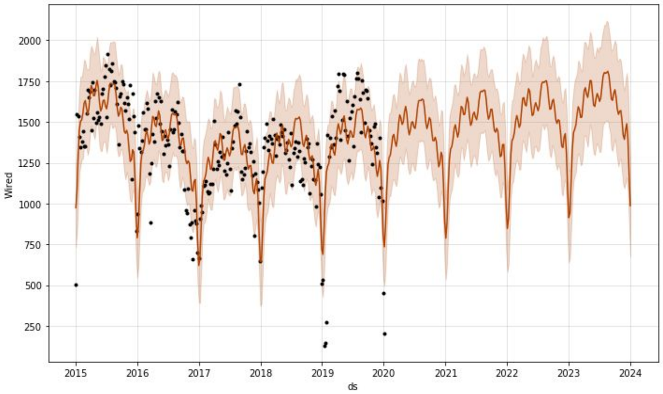
As with the previous method, I used it to forecast the year 2024 and compare their R^2 scores, but only compared data by MONTHLY and WEEKLY.

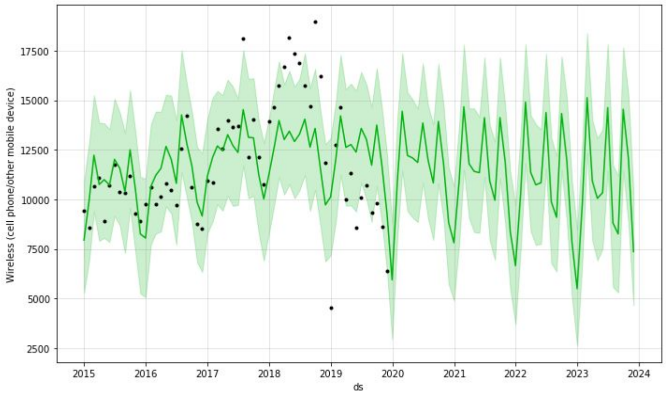
The results are provided in the following page, with the MONTHLY plots on the left and the WEEKLY plots on the right. Each method is color-coded as: Internet (VOIP), Wired, and Wireless (cell phone/other mobile devices).

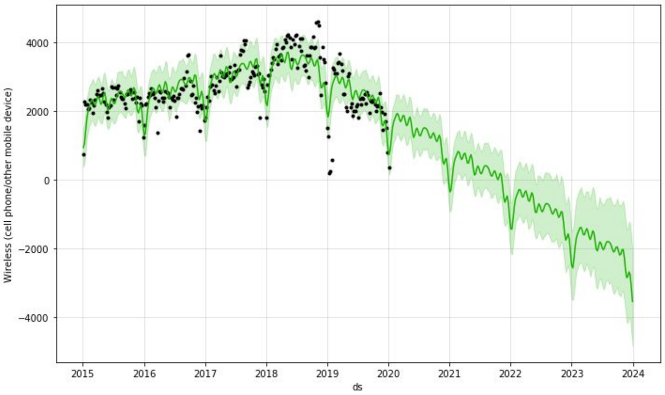












The resulting R^2 scores are shown below:

	MONTHLY	WEEKLY
Internet (VOIP)	0.6468	0.7082
Wired	0.6473	0.5690
Wireless (cell phone/other mobile device)	0.4033	0.6452

Clearly, the models provided by the FBProphet are much better predictors due to the API's robust algorithm, thus yielding much higher R^2 scores.

The forecasting plots show that the use of Internet methods will become higher and higher, while the use of Wired devices is more or less the same. However, for the Wireless, the Monthly plots and Weekly plots show different outcomes: the former shows it will neither decline or rise, while the latter shows a decline (although we cannot be certain if Wireless robocalls will cease).

Final Thoughts

As with other data analyses, this project aims to help the intended audience determine the appropriate action given the information gathered for this project. Given that there are endless possibilities regardless of a dataset we start with, there are more questions we can ask ourselves and/or address the existing questions differently; thus, it may or may not be the end. How the data is or will be used will be in the hands of the readers or the audience.