**Background**

This document is intended to be a complete walk-through of this project for those brand new to the baywheels data. This is data that Lyft, who runs the bike share program in San Fran, releases on a monthly basis as a csv file on this website: <https://s3.amazonaws.com/baywheels-data/index.html>

Note that Ford used to run the program before Lyft so that is why Ford’s name is on some of the files. Column names and detail of columns differ slightly between the Ford and Lyft csv files. For the purposes, of this project, we will only use fields contained in both Ford and Lyft datasets.

Here are the key fields

1. Start and end times
2. Start and end latitudes/longitudes (note it seems users can park bikes in locations outside of the bike stations so these lat/long are not just station locations
3. Member or non-member (there is a monthly/yearly membership to get discounted/free use on the bikeshare program. Google for more info if interested).
4. Bike Share for all indicator: This is a program that allows low income residents to get a heavily discounted membership for baywheels

**Goal**

The goal for this project was to predict future total daily ride time. This is for multiple reasons

1. To see if there is a trend or seasonality in ride time
2. To see if covid had any effects on ride time

**Order**

As you can see, there are 4 iPython files and 1 Tableau file on the GitHub. Here is the order of the Python files and a brief description of the purpose of the file. For more detailed descriptions, please look at the actual iPython file. Note I did not include the baywheels word in the file names below

1. data\_pipeline
   1. Download all zipped files and unzip csv
   2. Combine each of the 10+ csv files into 2 big files, one for Ford and one for Lyft
   3. Push to AWS S3
2. data\_cleaning
   1. Take a look at the fields in the Ford and Lyft data
   2. Determine what fields are useful to me
   3. Determine how to combine them, since both datasets have some different field and field names
3. EDA: Given the simplicity of the data, Tableau does a much better job of exploring it than this python file does and I recommend you look at that instead of this.
   1. Explore the data and get some interesting statistics - like ride count, avg duration, etc..
   2. See how our 2 variables user\_type and bike\_share\_for\_all\_trip affect the variables. (E.g. do different user types have different usage patterns)
   3. Examine usage based on time of day, time of month, time of year and overall trend over past 5 years of data
4. Modeling
   1. Get data into daily format
   2. Feed into fbprophet
   3. See how much MAE reduces from baseline model

Lastly, the Tableau file is great for understanding the data and is published here: https://public.tableau.com/app/profile/abhinav.gadde/viz/baywheels\_visualization/Totalview?publish=yes

**Reasons for using fbprophet**

After thinking through this, I have decided using fbprophet would be the best modeling choice. Here are my reasons for that instead of the more traditional ARIMA/SARIMA models.

1. Lots of different types of seasonality in my data that would make it hard for SARIMAX to catch
2. No need for data to be stationary
3. Automatically accounts for outlier year caused by covid (or very easy to tune a hyperparameter for this).

**Results**

I split my data into training and testing. The testing data is everything from 2022 and the training data is for everything prior to 2022.

As you can see from the modeling python file, the fbprophet model does decrease the MAE (mean absolute error) by about 13% (from 1.178M minutes to 1.027M minutes). Therefore, it does provide some value compared to the baseline model, which just assumed the t+1 value would be its preceding value at t. However, probably due to lack of hyper-parameter tuning, the lift isn’t as high as hoped for.

**Future improvements**

This is a minimum viable product, and many improvements / enhancements can be made in the future. The biggest ones are listed below

1. Make it so each month when a new csv file is put on the website, my script autopulls that and combines this with the existing data in the AWS S3.
2. Make my AWS S3 data public so anyone can easily access the cleaned combined data
3. Get hourly/daily weather and see how that impacts ride time
4. Connect Tableau directly to the AWS S3 data instead of using an offline copy so it can auto update when the S3 data updates
5. Tune the hyperparameters on the fbprophet model to reduce MAE

**Helpful links / resources**

How to upload/download data into/from an AWS S3 bucket: <https://towardsdatascience.com/how-to-upload-and-download-files-from-aws-s3-using-python-2022-4c9b787b15f2>

How to use fbprophet: <https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/>

How to get a baseline model: <https://machinelearningmastery.com/persistence-time-series-forecasting-with-python/#:~:text=Persistence%20Algorithm%20(the%20%E2%80%9Cnaive%E2%80%9D%20forecast)&text=The%20equivalent%20technique%20for%20use,step%20(t%2B1)>.