Bus Ridership Forecasting

C964 Computer Science Capstone

Joseph Belian

Western Governors University

Contents

Section A - Letter of Transmittal

To whom it may concern:

The COVID-19 pandemic has dealt a severe blow to public transportation usage. Though vehicle traffic is down in general, the industry is uniquely burdened: the tight quarters of busses and train cars, along with frequent interactions with members of the public, make for strong vectors of disease spread. This has strongly disincentivized would-be riders from riding on the company's bus lines.

To counteract the harsh fall in revenue, the Department of Research and Development has proposed mitigating the losses by prioritizing the more resilient lines, i.e., those lines that experienced a smaller decrease in ridership, and those that are elastic enough to bounce back from temporary dips. To facilitate this goal, the Department of R&D outlines the following: a data product will be developed that that takes ridership counts across each individual line and predicts what their future values will be. By using a pre-pandemic set of ridership numbers, we can predict what current ridership numbers should have looked like had the pandemic not occurred. We hypothesize that this will allow the business department to fine-tune their calculations for each bus line's value to the company.

The Chicago Transit Authority will provide the data, which will include the number of riders for each bus line, as well as data that displays the route of each bus line on a map. The data solution will be created by a team of two software engineers, one senior and one junior, using the Waterfall methodology, as the scope of the project does not appear to warrant the Agile approach favored by larger teams. A liaison to the relevant business department will also be required. The product will be accessed via the company intranet. During the data wrangling process, it is unlikely that the design goal will change significantly as information is discovered and processed; therefore, the project requirements

established under the Waterfall methodology should hold for the duration of development. The estimated funding needed for the project is $14,800 for the initial solution, while its long-term yearly maintenance costs is estimated at $100.

Many of the stakeholders are nervous about the company's short-term viability due the severe impact of the pandemic, even if many are also clear-eyed that this loss in profitability is a temporary dip. Nevertheless, the company actively taking steps to mitigate revenue loss will ease stakeholder anxiety.

The largest ethical concern for this product is that the data and algorithms may point towards the reduction or the elimination of certain bus lines and services. This will have a negative effect on local communities, especially communities with low incomes. Company bus lines may be the only feasible option for some of the company's customers. Thoughtful oversight is needed to consider this human cost before a seal of approval is given.

As for myself, I bring 30 years of software development experience and 20 years of leadership experience. I design both internal and consumer-facing products, with a specialization in traffic analytics. I have worked on industry standard products, some of which see over one million daily users. Software that I have worked on and led teams in is used in numerous federal and state agencies, including the Illinois Department of Transportation. I look forward to working with you.

Sincerely,

Joe Belian

Software Engineer

Section B – Executive Summary for IT Professionals

Problem Statement

The COVID-19 pandemic has resulted in severe declines in ridership across all forms of public transportation. By April 2020, public transit agencies collected enough data to determine that ridership levels decreased by 73% across the country (Qi et al., 2021). While the loss is staggering, it is possible for the company to mitigate the revenue loss and operational inefficiencies. A software product with predictive capabilities will be able to determine which bus lines have decreased the most by comparing bus line ridership during the pandemic versus their hypothetical values had ridership not been affected by the pandemic.

Currently, there is not enough data to make accurate predictions about future ridership by using the low pandemic numbers; however, once enough time passes, the forecasting model should be able to produce a meaningful trend line based on predictions from currently low ridership. Additionally, future analysis will be able to point out geographic imbalances, which could point to a recommendation for line consolidation.

Customers

The product is meant for internal use by public institutions and private businesses. The required input for data prediction and visualization is a CSV file formatted for the program's required parameters. As such, it requires minimal technical know-how on the part of the user so long as the data is clean. The user will then be able to see actual versus predicted ridership rates, which can be used in further analysis. Users will have access to the program portal via the company's intranet.

Existing System Analysis

The company is, unsurprisingly, unprepared for the severity and persistence of the pandemic. Moreover, the company does not employ machine learning for predictive modeling of bus ridership. This proposal will allow the company to right itself in troubling times by freeing employee hours spent on manual ridership analysis.

Data Availability

The data that will be used as a test case and is the one most relevant to the company's needs, will be provided by the Chicago Transit Authority. Data for ridership numbers goes back to the year 2001, though for the project's purposes only a few years of past data are necessary. Most agencies keep good records regarding public transportation; however, some of it may need cleaning, as there were a few incidences of row duplication in the CTA's provided CSV files. Any erroneous data that slipped by the agency's error handling will likely require a manual inspection for further cleaning.

The KML files provided by the CTA are used to display geographic information. These files are typically used in GIS software and web-based solutions like Google Maps. For the proposed data product, KML files will be used to visualize losses on a map of CTA bus lines.

https://www.transitchicago.com/data/

Methodology

The Waterfall methodology will be employed during software development. The team size will be small, consisting of a senior and junior software developer, and the stakeholder investment and communication requirements will be minimal. For these reasons, while usually an Agile methodology is

preferred, it benefits most from a large development team that need high levels of communication across many moving parts. The requirements for this project will be set by the developers with input from the relevant business departments as to what data must be gleaned.

Project Deliverables

- This document.
- A scope document.
- A project requirement document which has been agreed upon by the relevant stakeholders.
- A milestone schedule that ensures the project is progressing appropriately.
- A test plan detailing compliance testing, unit testing, and integration testing.

Product Deliverables

- A machine-learning predictive model that will read in ridership rates for comparison and analysis and can be developed and adapted for future needs.
- A friendly GUI that can be presented to non-technical persons and stakeholders for future project needs.
- Security measures will require a token-based password system for two-factor authentication.
- Logging of access to the data product as well as actions such as CSV files uploaded or downloaded.
- Architecture recommendation and documentation for running and delivering the product across company intranet.

Implementation Plan

1. Collect raw data of ridership statistics from government agencies and private transportation companies.
2. Clean the data.
3. Designate the most recent year of ridership statistics as a dataset for testing, then designate the previous years as a dataset for training the predictive model. Utilize descriptive techniques to verify that model fits without overfitting.
4. Develop the predictive machine learning model with a non-descriptive technique: currently, the implementation recommends the Prophet model developed by Facebook, as it handles seasonal data well.
5. Develop a portal to read in CSV data and perform predictive analysis on it with the model.
6. Dock the portal to a cloud-based intranet software provider.
7. Deploy to production and perform acceptance testing.

Evaluation Plan

After the implementation plan is complete, a case study of the Chicago metropolitan area will be performed. An evaluation of the predictive model's efficacy in determining how bus lines should be operationally prioritized will take place. If the model is determined a success, transit data from similarly sized cities will be used for further benchmarking.

While the usefulness of the model to the company is yet to be determined, the accuracy of the predictive model itself will primarily be an object measurement using the mean absolute percentage error, or MAPE. Against good data, an effective model will train its algorithm on several years of training

data. Then, its predictive capabilities will be tested against a year of test data. The closer the MAPE value hews to zero, the more accurate the prediction.

Programming Environment and Costs

Labor is expected to take up to one and a half months, including coordination with relevant state agencies and data procurement, as well as completing the implementation and evaluation plans.

Initial labor cost: $42/hour or $13,440 over the course of 6 weeks
Yearly maintenance: $100
Estimated cost of workstation upgrades: $1,300

Hardware:

- Windows 10 Pro
- Hardware: Intel i5-4690K CPU
- 16 GB RAM

Software and Notable Packages:

- Jupyter (prototyping only)
- PyCharm IDE
- Python 3.9.9
- geopandas
- libpysal
- mapclassify
- matplotlib
- numpy
- pandas
- plotly_express
- seaborn
- statsmodel

Timeline

Timeline will follow the usual schedule of a Waterfall methodology.

Milestone	Date
System Requirements	Jan 2 – Jan 4
Analysis of Requirements	Jan 7 – Jan 9
Design	Jan 12 – Jan 15
Deadline for data procurement	Jan 15
Coding	Jan 17 – Jan 28
Testing	Feb 2 – Feb 7
Deploy	Feb 14

Section C

Data Methods

Descriptive method: Multiple graphs in the Jupyter Notebook display ridership over time in both total numbers and separated by individual bus lines. A histogram displays the number of bus lines that experienced a certain percentage loss of ridership between predicted and actual values.

Non-descriptive method: Prophet, a procedure for forecasting time series data was used as the non-descriptive method. Prophet was chosen for strength in time series predictive modeling, as it works exceptionally well with time series that have strong seasonality to them, as transit data does.

Datasets

The datasets come from the Chicago Transit Authority's website.

https://www.transitchicago.com/data/

Analytics

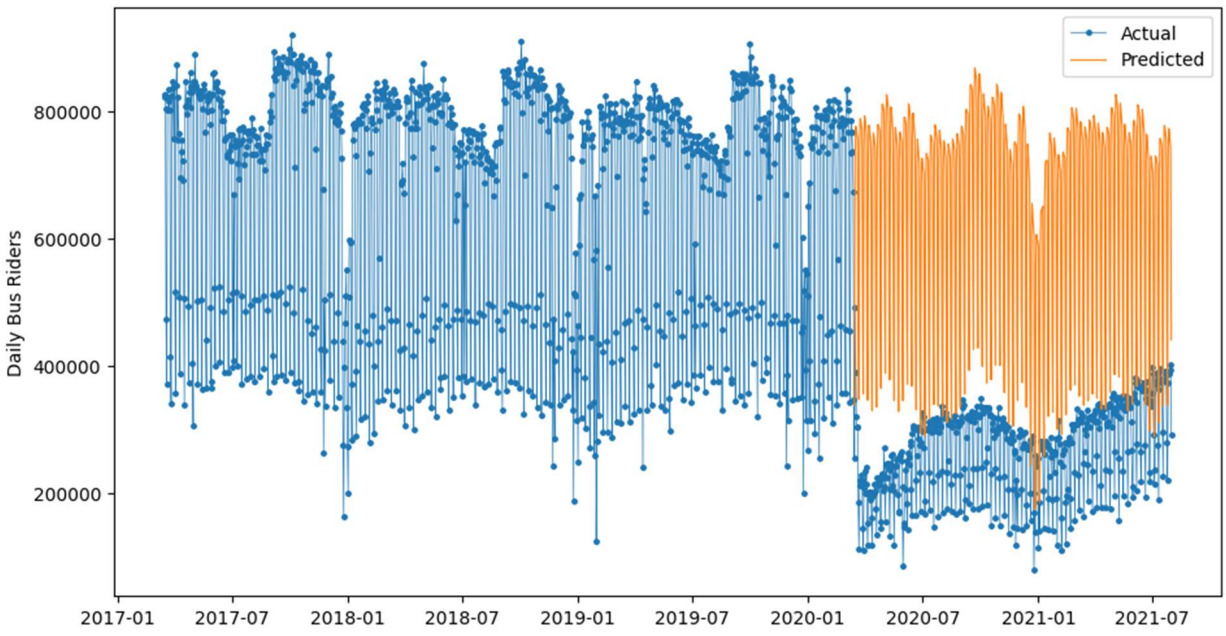
With this data, time series forecasting of bus lines can accurately predict the expected ridership of individual and total bus lines. Depending on geographical and socio-economic factors, different lines may bounce back quicker to their normal levels, while some lines may hew closer to their new lows. Further analysis can be performed by plotting bus lines' geographic placement against U.S. census data to determine which factors are strongly correlated with the elasticity of a bus line.

Data Cleaning

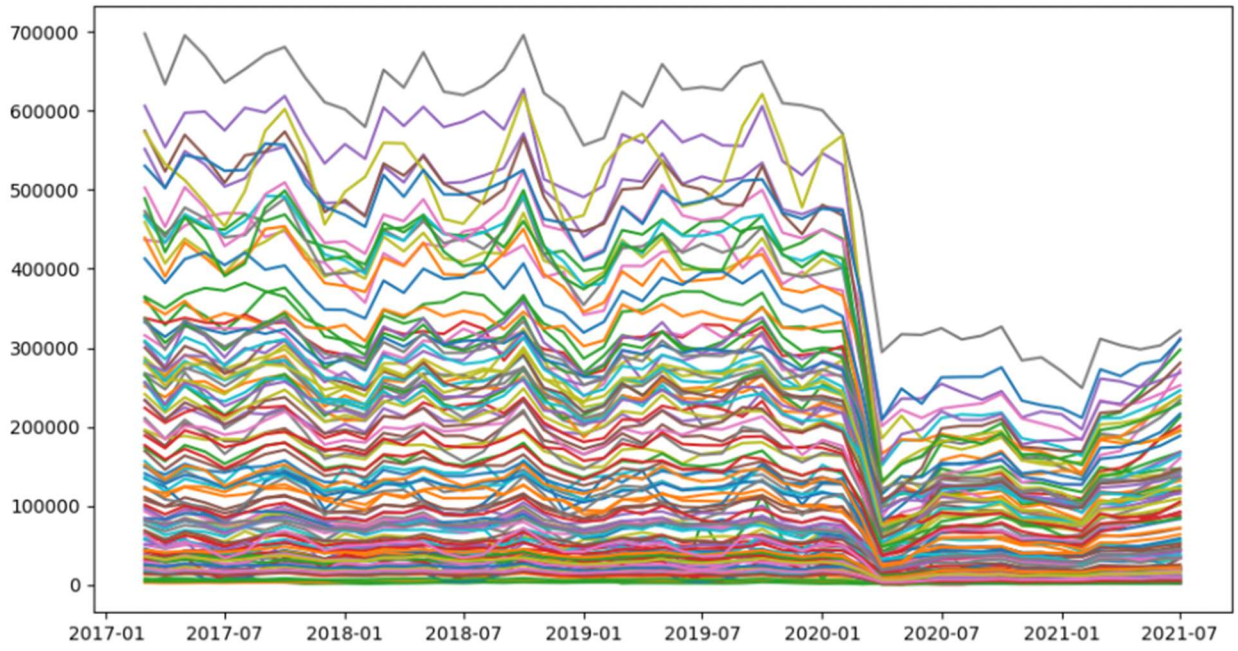
Irrelevant column data were dropped, while duplicate row entries were sorted out by selecting for non-unique values.

Data Visualization

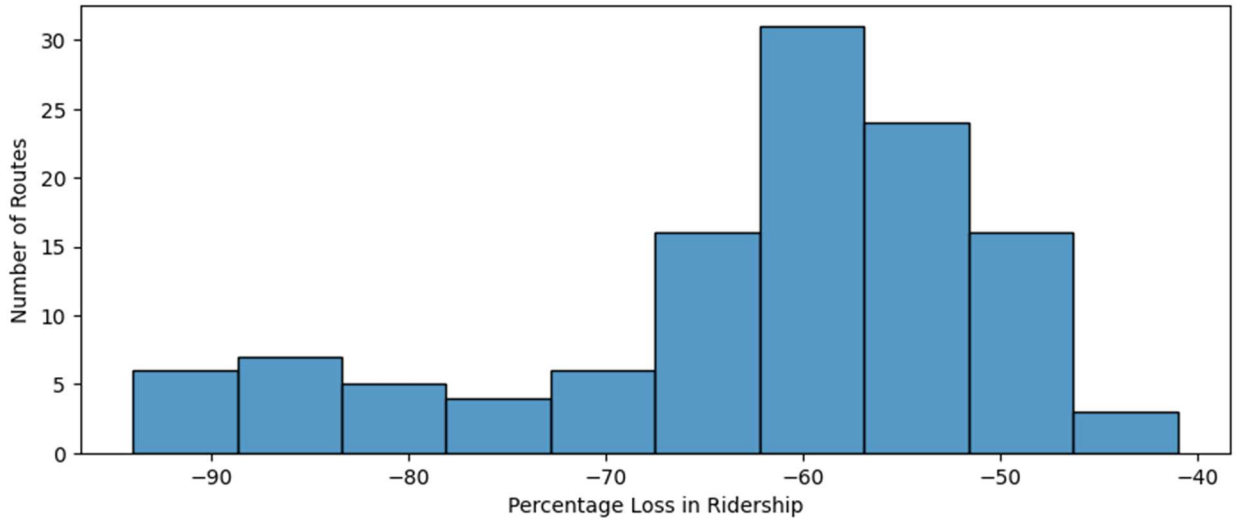
The JupyterLab prototype displays three of the four visualization types expected for the project. Four graphs on the prototype display ridership over a time series graph. Two of those graphs also show predicted values concurrent to actual values.



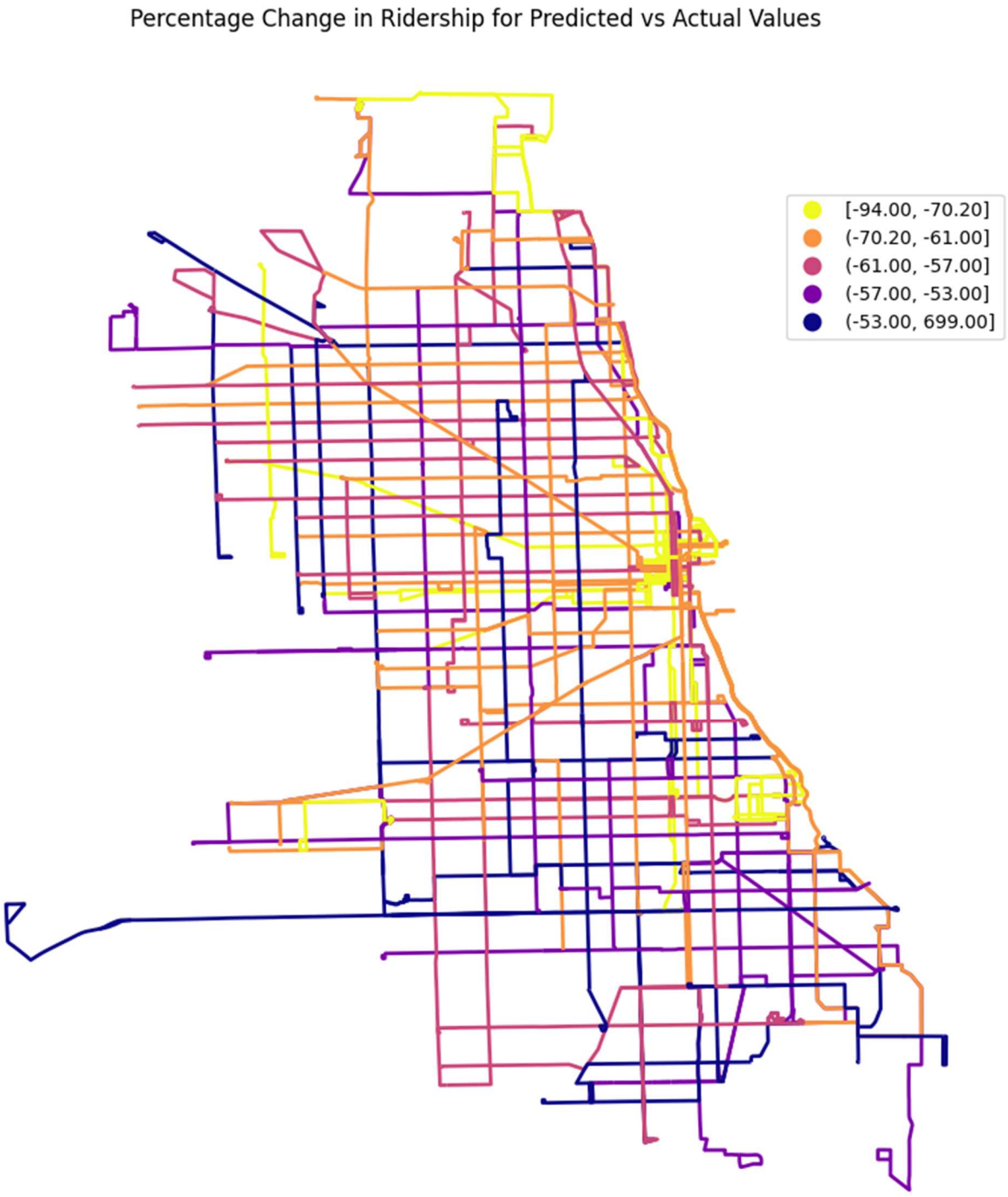
One graph separates the time series into every individual route – while it is not particular readable nor informative, it is useful as an illustration of how parallel the lines are. Initially, there was hope that a visual would show if certain lines would cluster together; however, that proved to be for naught. There is too little differentiation between the lines for clustering.



A histogram displays the number of bus lines that experienced a certain percentage loss of ridership between predicted and actual values.



A useful visualization of the geographic clustering of ridership drop-off rates between predicted and actual values. The downtown of Chicago experienced losses into the 90% range. Other trends emerge: the south side of Chicago, in cooler colors, has more resilient ridership numbers compared to the north, in slightly warmer colors.



Real-Time Queries

On a local machine, the Prophet package provide instantaneous predictive modeling based on chosen parameters. However, a persistent bug prevents the package from installing correctly on Jupyter Notebook and adjacent services such as Kaggle. As a stop-gap measure, the data modeling was run on a local machine, and the saved results were uploaded as CSV files to the notebook. Effectively, this is invisible to the end-user, as the data produced will be the same.

Additionally, for further GUI manipulation, matplotlib graphs are fully functional widgets. The user can pan and zoom for each one.

Machine Learning

The product implements time series forecasting for non-linear trends and strong seasonality. The accuracy of the predictive model is weighed against its mean absolute percentage error, or MAPE.

Accuracy Evaluation

Accuracy of the predictive model is primarily weighed against its MAPE rating, which is a ratio that describes how far away the predicted value is from the actual value. The lower the ratio, the more accurate the model. When testing the 2019-2020 year using the years 2016-2019 as a training set, the MAPE value is 10.247%.

$$\mathrm{MAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

Security

Security measures will require a token-based password system for two-factor authentication when accessing the software portal, as this is a universal procedure across the company. The user will be able to manipulative the graphs and other data for visual purposes; however, this will not save to disk. The user will be able to download a CSV of desired data, and they will also be able to upload CSV data for modeling. Security checks will be in place to prevent malicious code from being spread through this channel.

Product Health Monitoring

The largest point of error in this software will likely be user uploads. Proper exception handlers will handle the majority of potential pitfalls. Regardless, a logging system will be implemented with current IT standards.

Dashboard

The dashboard is built with Python code in JupyterLab, then hosted in an executable environment via Binder.

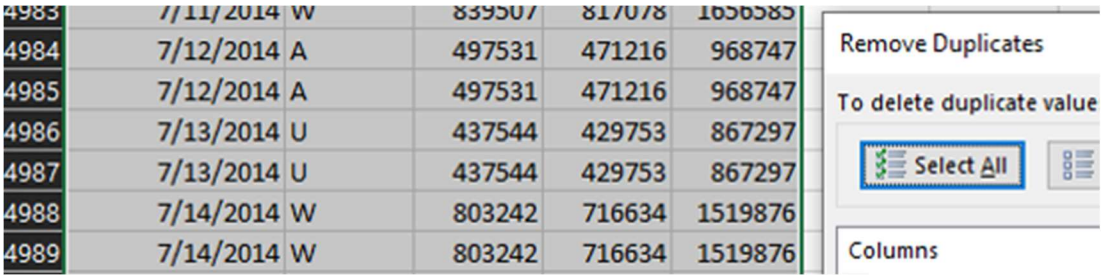
Section D

Business Requirements

The purpose of the project was to create a software environment that easily visualizes time series and forecasted data. The product will display ridership disparities in predicted versus actual values across all of the different routes available in a transit system. Additionally, the product serves as a starting point and foundation for data gathering and analysis. As the pandemic progresses, additional data will lead to additional insights: namely, that the currently low ridership numbers will have enough data to accurately predict future ridership.

Datasets

The raw CSV files from the Chicago Transit Authority were in good shape; however, some months were unsorted, and there were a few instances of duplicate rows. As this is a quick fix in Excel, the data was sorted with the auto-sort button, and then duplicate rows were removed with the "remove duplicate rows" button. If many more CSVs were being handled, this process would be automated.



4974	7/12/2014	A	497531	471216	968747
4975	7/13/2014	U	437544	429753	867297
4976	7/14/2014	W	803242	716634	1519876

Data Product Code

After the raw data has been cleaned, the forecasting model is called on. For the "total" boarding numbers, every day exists, so weekly and daily seasonality are set as True. For the individual routes, only the first of each month is available, so weekly and daily seasonality are set to False. The predictive model that Prophet creates is spit out as a dataframe with a number of useful features, but the most important is simply the Y value – the number of riders – which is then used for plotting on the graphs.

```
def prophet_forecast(dataframe, pred_begin, frequency):
    train_begin = pred_begin-relativedelta(years=3)

    df = dataframe.reset_index()
    df.columns = ['ds', 'y']
    df['ds']= to_datetime(df['ds'])

    train = df.set_index('ds').truncate(before=train_begin, after=pred_begin)
                        .reset_index()

    if frequency == 'M' or frequency == 'MS':
        model = Prophet(yearly_seasonality=True,weekly_seasonality=False,
                        daily_seasonality=False)
    else:
        model = Prophet(yearly_seasonality=True,weekly_seasonality=True,
                        daily_seasonality=True)
        print('Training period:   ',train_begin.date(),' - ',pred_begin.date())
        print('Prediction period: ',pred_begin.date(),' - ',
                        (pred_begin+relativedelta(years=1)).date())

    model.fit(train)

    pred = list()
    for date in pd.date_range(start=pred_begin, end=graph_end, freq=frequency):
        pred.append([date.strftime("%Y-%m-%d")])
    pred = pd.DataFrame(pred)
    pred.columns = ['ds']
    pred['ds'] = to_datetime(pred['ds'])

    forecast = model.predict(pred)

    return forecast, df
```

Hypothesis Verification

The hypothesis is that this kind of modeling would be useful data for evaluating a bus line's profitability. The reality is that any information gleaned in this scenario would be only marginally more useful than a simple year-over-year percent change. The original hypothesis was that clustering routes based on their individual trend lines would lead to some insight in correlative data, such as geographic clustering. However, this too proved difficult to glean information out of, as any attempt at fitting would overfit the data as all the routes are so tightly packed. It seems likely that once more time passes, a useful forecasting model could be applied to the real ridership numbers of the pandemic period. As for the current hypothesis that this kind of modeling is useful: while it is not bad data or information, its usefulness regarding business decisions is dubious.

Effective Visualizations and Reporting

Data visualization covered in depth in Part C, Data Visualization.

Accuracy Analysis

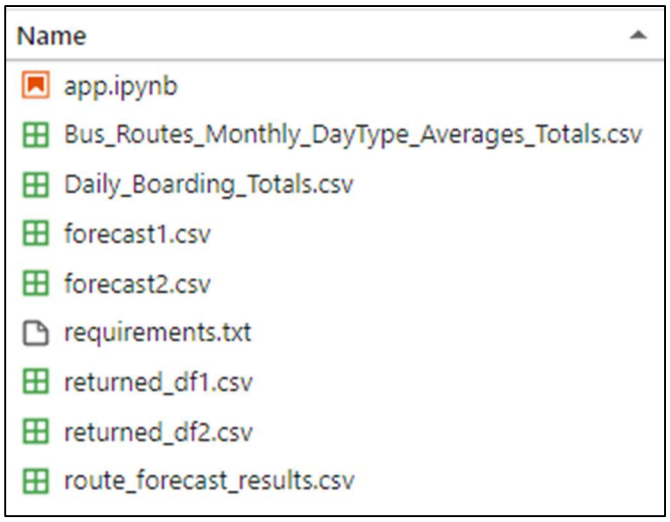
The data is very accurately presented. Considering the strong seasonality of transit ridership, the very consistent time series is not surprising. Accuracy for the "testing" sets based on "training" sets reaches a MAPE score of 10%, which is good.

Application Testing

Unit testing was performed in order to ensure that individual functions parsed well and worked correctly before moving on to the more decorous elements of the project, like the graphs.

Application files

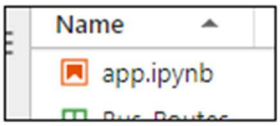
App.ipynb	The app and its main entry
Bus_Routes_... Daily_Boarding_...	Cleaned raw data for ridership values
forecast1, forecast2, returned_df1, returned_df2, route_forecast_results	Backup CSVs containing Prophet results
Requirements.txt	Packages and dependencies needed to launch the app



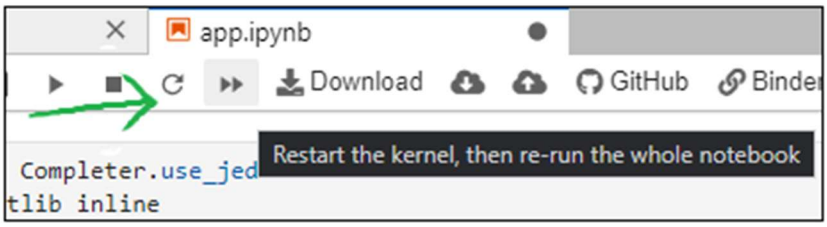
User Guide

Click this link: https://mybinder.org/v2/gh/jbelian/C964-Bus-Ridership-Forecasting.git/HEAD

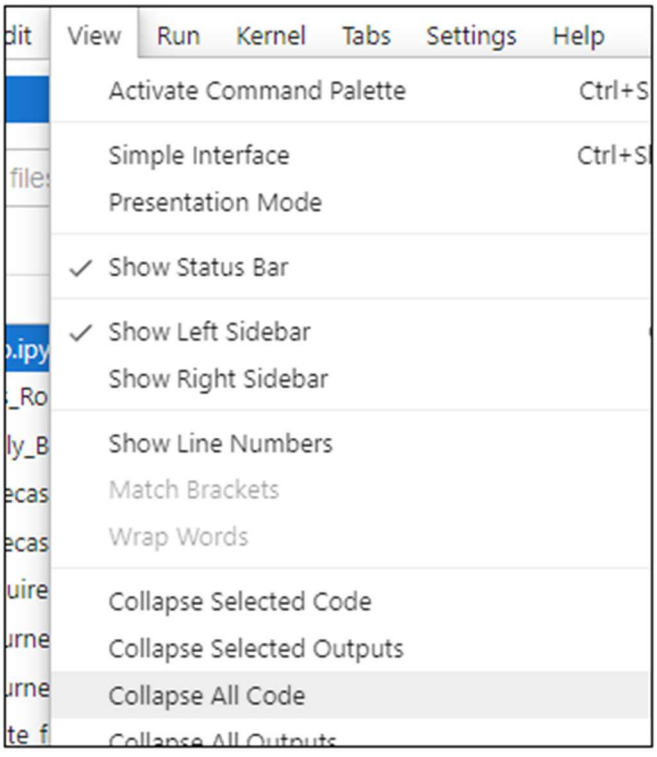
Double click app.ipynb



Restart the notebook if necessary



Collapse all code for a more natural dashboard if desired



Summation of Learning Experience

The one thing about my prior knowledge that would probably set me apart from others that have completed this task is basic Geographic Information Systems knowledge, which involves things like shape files and geographic coordinate systems. Originally, my approach for this project was going to involve geographical clustering based on bus ridership activity, but that soon proved to be out of scope for this project. The sole remnant piece of code from that attempt is the map of Chicago bus routes that I included on page 12.

This dataset proved difficult to develop a workable machine learning model for; however, I am satisfied with the solution and the presentation to the hypothetical posed earlier. Given that there is not enough data from the relatively short time the pandemic has affected bus ridership, the business case for this project is dubious; however, I believe it is a useful foundation for future modeling. Once ridership rates start to rise, it will be interesting to note if certain routes proved to be more resilient to the initial decline and more elastic to shifting incentives. With more data, I believe any attempts at finding correlations via geographic clustering and US census data will prove to be legitimately useful as a business case.

I did not have any machine learning experience prior to this project, and it took several attempts before settling on what data and what time series forecasting model I would use. The point of this project was, in any case, for the sake of my learning. While it was frustrating to feel like I tossed aside good work, I know that ultimately I created a better project for it. I can finally say that I am excited for having done the hard work to complete this degree and to move on to the next chapter of my life.

Section E

Yi Qi, Jinli Liu, Tao Tao, Qun Zhao,

Impacts of COVID-19 on public transit ridership,

International Journal of Transportation Science and Technology,

2021,

ISSN 2046-0430,

https://doi.org/10.1016/j.ijtst.2021.11.003.

(https://www.sciencedirect.com/science/article/pii/S204604302100085X)