Manuscript Title

This manuscript (permalink) was automatically generated from banyekalaok/cee498report@9be4572 on December 4, 2020.

Authors

- **Ruihao (Robert) Zhang**
  XXXX-XXXX-XXXX-XXXX · banyekalaok · banyekalaok
  Department of Civil Engineering, University of Illinois, Champaign-Urbana

Chapter 1. Introduction

As part of the course project for the class, project group 7 acquired bus ridership per trip weekday data (from here on referred to as **bus data**) for the month of August from the Champaign-Urbana MTD bus organization (CU-MTD). To supplement the bus data, CU-MTD provided a manual that explains the process and methodology used to track, monitor and acquire the bus data.

Project Group 7's Objective

Group 7's project objective is to use the bus data to predict the buses load averages (average number of passengers onboard during a trip). This project is of particular interest to CU-MTD because in 2020, the bus load averages have significantly decreased due to the COVID-19 pandemic. Therefore, any results or conclusions that help CU-MTD better predict the load averages could increase their operating efficiencies and minimize costs.

Full Report Structure

The report was developed using Manubot to allow for a collaborative effort among the 6 group members. The explaratory data analysis (EDA) and model were developed using Python 3 in a Kaggle Notebook and Jupyter integrated development environment (IDE). The content in the following report is broken down into the following 4 main chapters:

1. **Introduction** - The current chapter which introduces the project scope and objective.
2. ** Explaratory Data Analysis** - This chapter details the EDA process, findings and key takeaways.
3. **Model Development** - This section describes the step-by-step methodology used to determine the best model to predict the load averages.
4. **Conclusions** - This section briefly highlights the key takeaways from the model development efforts and the report in general.

Chapter 2. Explaratory Data Analysis

The content in chapter 2 is broken down into the following 3 sections:

1. **Data Tidying** - This section discusses how the raw bus data was cleaned and prepared for the EDA analysis. The section also details the independent and dependent variables used in the analysis.

2. **Investigating Data Insights** - This section details the bus data EDA process and outcomes.

3. **EDA Summary and Conclusions** - This section summarizes the key findings and conclusions from the EDA.

Data Tidying

Independent Variables

The CU-MTD bus data was provided as a csv file. Therefore, the easiest way to setup, tidy and initially analyze the data was in tabular form. The bus data was then assessed for meaningful ways it can be subdivided based on identifying independent variables. As shown in figure 1 below, the number of trips are fairly evenly distributed throughout the week.

Figure 1. Trip distribution per weekday

Notice that Monday, the start of the work week, has the most trips. Since we only have 1 month of data (August), no strong conclusions can be made at this point. The following section summarizes the EDA results that are relevant to the model development effort.

Of the 48 bus lines figure 3 shows that the first 11 lines below (indicated by the boolean output “True”) conducted more than 1,050 trips in August. 1,050 trips represents an average of 10 trips per workday, which is considered substantive in this analysis.

Figure 3. Bus line distribution

Dependent Variables

One of the unique challenges faced in this data was what to do with the time data (schedule start and end times). Firstly, the data was provided as strings, therefore, they needed to be converted to datetime format. Secondly, some of the data entries that were not on the conventional 24-hr time format, i.e., sometimes were between 24:00 and 26:00 hours. This is likely because the timestamps represent the bus driver workshifts. Workshifts are easier to monitor and track on a continous scale from clock-in to clock-out than to break-up because of the start of a new day. A function was created to correct the time to be in the 24-hr format then the times were used to determine the duration of a trip (schedule end time - schedule start time).

Investigating Data Inisghts

Show meaningful plots and correlations. A few examples are provided below

As can be seen in figure 8, each day has a very similar distribution, however, Monday and Thursday have some outliers beyond the maximum. This gives some insight and confirms that the number of trips per day are fairly evenly distributed.

Figure 8. Box plot of duration per day full dataset.

Figure 14 informs that the a $ trip < 12 minutes $ long have no stops. The number of stops then steadily increase with time to a max of 42 at a trip of duration ~ 55 to 65 minutes.

Figure 14. Bar plot of duration against P-stops filtered dataset.

EDA Results Summary

The following section summarizes the EDA results that are relevant to the model development effort. Through the EDA of the bus data the following information was noted about the data set:

- The data set only includes weekday trips and the data is fairly evenly distributed.
- “Line” is the name of the bus line, which is what most riders are familiar with. There are 38 unique bus lines and 11 of them that do more than 10 trips per workday.
- The data has 6 categorical features – with the exception of pattern, the other 6 features will be used in the model development effort.
- 24 numerical features – the following features wil be used in the model: P-stops, total in and total out (flux?), and PM.
- The label is the load avg.

The following features were identified to have strong predictive ability, indicated by a high correlation with the label (correlation in parenthesis): * P-stops (0.61). * Total in (0.78) and total out (0.78). * Max (0.90). * PM (0.85).

Chapter 3. Model Development

The content in chapter 3 is broken down into the following 2 sections:

1. **Data Setup and Feature Engineering** - This section details the pre-processing that took place prior to model development.
2. **Model Development** - This section describes the step-by-step methodology used to determine the best model to predict the load averages.

Data Setup and Feature Engineering

Similar steps are taken in this section as were taken in the EDA process. However, the overall goal here was to clean the data in such a way that it produces informative, predictive models with minimal bias and over- or under-fitting. Therefore:

- The data field was converted a day of the week.
- The bus lines can be cleaned so that they match a format familiar to riders.
- The start and end times are converted such that they reflect times and the difference between the start and end time is stored in the data sets as a **‘duration’**.

The following paragraphs describe the particular ways the training data was cleaned to result in meaningful predictive models. The 2 main ways feature engineering will be conducted on this data set are: 1) The developers knowledge on the dataset (as presented in the EDA), and 2) assessing the statistics on the dataset.

Based on the EDA, we know that the following features have no value to the model we’ll be developing. Therefore, the following features can be removed from the data set:

- **Duty**, **EMPTY_1**, **EMPTY_2**, **EMPTY_3** and **Graphic** - Features with blank entries and no description in the manual on the data set.
- **Capacity**, **Full capacity**, **Capacity (pract.)**, **Load factor [%]**, **Load factor (pract.)[%]** and the **PM factor [%]** columns were all 0's.

The EDA also revealed that the following data sets had numbers that didn't actually correspond to a numerical value (i.e., the numbers are codes that have no value to the model development effort).

- **No (train)/index (test)**, **Trip**,**Block**, **Course**, **Pattern**, and **Vehicle**.
- **Date**, **Sched. start**, and **Sched. end** - They were used to determine the schedules, day and duration. From here on, they no longer have model development value.

Feature Scaling

The purpose of feature scaling is to transform the data into common scales.

Before scaling the data, lets view some data statistics to refresh our memory on what was learned in the EDA. The key takeaway from the graphics below are:

- A linear model might project the load avg but it likely won't be the best model.
- The correlations matrix confirms that the identified predictive features are likely to result in a good model (except for minimum).

Figure #. Comparison of key predictive variables against load average.

Figure #. Correlation matrix of key predictive variables.

From the statistics above, notice that the minimum load (**Min**) has a very low correlation with the load average, therefore, min has low predictive value. Also, notice that **total in** and **total out** are strongly correlated, implying that the sum of boardings during a trip is almost always the same as the sum of alightings. Therefore, only 1 of these 2 features are needed. The **total out** will not be considered in the model development effort.

Scaling Numerical Features

Models learn best and fasted when the data is scaled, most commonly between values of -1 and 1 or 0 and 1. Since, all the numerical values are greater than 0, the numerical features were scaled using normalization (scaling from 0 to 1).

Converting Categorical Features to Numerical

To allow the model to learn from the categorical features, they need to be converted to numerical features. The following section details how the conversions were conducted.

- The EDA revealed that there are 39 unique lines that conduct a wide range of rides from 2 to 1,300. Since the line is a strong predicator of bus loads (i.e., popular lines will likely have larger loads), the line were converted via one-hot encoding.
- The EDA revealed that there are 7 unique vehicle types that conduct a wide range of rides from 452 to 9,414 . Since the vehicle type is a strong predicator of bus loads (i.e., larger vehicles can carry larger loads), the vehicle type was also converted via one-hot encoding.
- Similar logic as above was used to convert the week days using one-hot encoding.

The last step in the setup process was to split the training data into training and validation data. This step is done to minimize the likelihood of creating a model that will over or underfit the data. A split of 70% and 30% between training and validation was used.

Model Development Process

The follow sections goes through a step by step process to develop and identify the best model to predict the load average on the test data.

** I recommend we show 3 models. For each we should: discuss parameters and hyperparameters, show RMSE vs epochs (and perhaps 1 other metric [I used MAE]), then show predictions against targets.

Establish a Baseline Model

Before building a trainable model we'll create a baseline that simply returns the load average.

As expected, the baseline model had a high root mean square error (RMSE) and mean absolute percentage error (MAPE) on both the training and validation data. The next section assesses if a dense model can produce a better model with lower metrics.

Develop a More Complex Model

As can be seen, the dense linear model results in better (lower) metrics than the baseline model. However, the improvement in the model performance is not that significant. The next section assesses if a dense non-linear model can produce a better model with lower metrics.

Most Predictive Model

** My model was best with relu layers and dropout regularization only. Would probably be even better if it had automatic hyperparameter and parameter selection!**

Chapter 4. Conclusions

As was shown in chapter 3 the most predictive model was able to predict the load average with ## accuracy. Some factors need to be taken into consdieration about this process:

1. A significant amount of pre-processing was used - To make this model more cost- and time-effective the input data needs to be tidy before going into the data. This can be achieved by changing the way the data is stored or creating classes and functions that automatically clean the data.
2. Reliablity - This model was developed using data from a single month from 2020 (August). Therefore, although the model had high predictive potential with the August data. It is possible that the model is not able to accurately predict load averages for other months were passenger habits may significantly differ (e.g., middle of summer, winter or spring). Thererfore, to truly optimize the model a larger year round data set is recommended. It would probably be best to train the model using at least 1 month of data from the 4 seasons of a year.
3.

References