Development of Transit Ridership Prediction Models for BMTC

A Project Report submitted in partial fulfillment of the requirements for the degree of
Master of Management (M.Mgt)

by
Revanth Guthala
under the guidance of
Prof. Parthasarathy Ramachandran

Indian Institute of Science
Bangalore

Department of Management Studies (DoMS),
Indian Institute of Science (IISc),
Bangalore - 560012, India, June, 2019.

Acknowledgements

I would like to express my sincere gratitude and deep appreciation to my project guide **Prof. Parthasarathy Ramachandran**. This project work wouldn't have been possible without his support. He left no question unanswered, and instilled me with the focus, determination, and knowledge necessary to complete this project.

I take this opportunity to thank my examiner, **M.H. Bala Subrahmanya** for his timely insights and critical examination of my work.

My special thanks to the faculty of the Department of Management Studies, IISc, who have shared valuable inputs during our course duration.

I thank my industry guides **Ipsita banerjee** and, who has been a constant pillar of support and encouragement and **Prof.Abdul Rawoof Pinjari**, whose valuable inputs, shaped my thesis work. Their suggestions with their domain expertise have been critical during this project.

I would also like to thank
Centre for infrastructure, Sustainable Transportation and Urban Planning
Indian Institute of Science (IISc), Bangalore

for giving me the opportunity to work on such an interesting problem.

I would like to thank my brother Hemanth Guthala,research associate in Indian School of Business Hyderabad who helped me in analyzing spatial data and suggesting the tools to make the work fast & efficient.I am faithful to my friends Monica,Nikhil,Rohit and others for their continued love and support throughout the duration of my Masters Studies.Finally, I would like to dedicate this journey to my parents Sarath Kumar and Vijaya Nirmala for encouraging me in every aspect in my life.

Contents

List of figures

Abstract

The BMTC is one of the three Subsidiaries of Karnataka State Road Transport Corporation, in the context of the city's expansion in the year 1997.The BMTC introduced its Intelligent Transport System (ITS) in 2016, with three main functions as part of it – electronic ticketing, vehicle tracking and public display of information.As part of the ITS, the entire ticketing system of the BMTC got overhauled with new electronic ticketing machines (ETMs) being deployed. The existing ETMs could only print tickets and log the data of sales. This data is fed into servers and made the path to various analysing levels with solving problems such as traffic,ridership and revenue etc.

Bengaluru region is included in the analysis, representing different types of communities. This research aims to better understand the relative and combined influence of transit service characteristics and urban form on transit ridership at the stop level. We use stop level ridership data from 5000 bus stops in the Bengaluru region, Ridership at these stop levels as the dependent variable for regression and other models.Categories of independent variables tested include: (1) socio-demographics; (2) transit service characteristics (e.g. frequency, hours of service etc.); (3) land use (employment, population, land use type, pedestrian destinations, etc.); and (4) transportation system (e.g. street connectivity, bike lanes, etc.). The final model results indicate that the model does a better job explaining the variation in ridership at the stop-level; the adjusted-R2 12 is 0.64, compared to 0.67 for the machine learning models, and 2% MAPE for the time series models.

Chapter 1

Introduction

The most important of electronic ticketing; not because it makes things easier for the conductor or passenger, but because it results in a gold mine of data,that can go a long way in solving the city's transport problems.Ticketing data in-principle is used by transport corporations (transcos) to identify how many tickets were sold, how many passengers boarded, and what denomination tickets they purchased. In the older, punched ticket system used by BMTC once tickets were sold, conductors would log the serial number of the ticket on top of the bundle at fixed intervals. The difference in serial numbers would indicate the number of tickets sold. However, data such which stage the passenger travelled would be unavailable. Two passengers travelling the same distance but at different points would pay the same.
E-ticketing solves that issue to a great extent. It logs the bus stop where the passenger boarded and disembarked, how many passengers travelled on which stage and more. Further, with smart cards, even passes can be tracked, thus giving the operator a clearer picture of the total passenger count.

In this chapter, the motivation behind this project is stated in Sect. 1.1. Sect. 1.2 discusses the existing and proposed processes. The objective and the modules are discussed in Sect. 1.3 and Sect. 1.4, respectively. The project organization is given in Sect. 1.5.

1.1 Motivation

There's a clear link between growing ridership and overhauling bus service. In almost every Indian city, bus service carries the majority of trips, so it should be no surprise that cities have to improve bus service to grow ridership.

The objectives of this discussion paper are to:

1. Provide a high-level overview of the potential uses of Transit ITS data for planning and management purposes,

2. Identify the various challenges in using the data, and

3. Recommend research and other initiatives that would enable transit agencies to make more effective use of the data, and position the transit industry for a future of ubiquitous data and data-driven decision-making.

Phase I	Phase II
Development of statistical ridership model for BMTC Dec 2017 & Jan 2018	Transit Boardings Estimation Tool Similar to Public transit agencies in Florida, USA. https://tbest.org/ (Polzin et al., 2011) CiSTUP plans to develop such a software tool for use by BMTC and other stakeholders.

The whole project is divided into two phases. My research will be much towards Phase I where I need to develop a model for ridership prediction on the available data of Dec 2017 & jan 2018. The study and decisions will be resourceful for creating the estimation tool or dashboard like structure.

1.2 Primary Sources of Transit ITS Data (Including the implementation of smart cards in BMTC)

There are three primary sources of data that are directly pertinent to transit planning and management.

• *Computer-Assisted Dispatch / Automatic Vehicle Location (CAD / AVL)*

The CAD/AVL system is the heart of most Transit ITS deployments. It continuously tracks all transit vehicles in real-time, which enables efficient and effective operational control, incident management, security response, and service restoration. By comparing the real location of vehicles to their scheduled location, it enables continuous monitoring of schedule adherence. This can then be used to calculate Estimated Time of Arrival (ETA) of vehicles at all stops downstream and thus drive real-time information at displays at stops, on the internet, on mobile devices, etc. But more pertinent to this Discussion Paper, CAD/AVL provides a wealth of data from on-board devices (e.g. location, door opening sensors, odometer, etc.) that is geo-coded and / or time-stamped describing what the transit vehicles are doing. This in turn can be transformed into information on schedule adherence and On-Time Performance (OTP), running times, dwell times, delays, vehicle speeds, etc. It is important to recognize that it is not only just the GPS location that is important, but that the monitoring compares real-time outcomes to schedules. It should be noted that some systems provide a very detailed second-by-second log of events / locations of a bus between pull-out and pull-in. These files are extremely large with a complex file format, and it can be a challenge to extract the specific data required for a specific type of analysis. In addition, CAD/AVL systems are typically used to capture information from a number of other on-board sensors (e.g. passenger counters, wheelchair ramp, bicycle rack, etc.) that can also provide valuable information.

• *Automatic Passenger Counting (APC)*

Automatic Passenger Counting systems can be developed as stand-alone systems, or integrated into AVL systems. A standalone APC system typically records passenger boardings and alightings with time and location coordinates. The collected data then is then off-loaded from the bus and matched to the stops and schedule database. After matching of the data to stops is completed, APC data can provide detailed profiles of

customer activity by stop and time of day, as well as accurate estimates of passenger loads. This provides a wealth of information on customer demand. Some systems do calculate loads in real-time while the bus is in operation. However, imbalances between on counts and off counts can lead to escalating load estimates, and this can be a bit misleading if real-time loads are being broadcast in passenger info systems.

• *Advanced Fare Collection (AFC)*

Data from legacy fare collection systems was typically of limited value because of its limitations. Data would be collected at turnstiles only on a periodic basis and provided aggregate total entries, etc. Electronic fareboxes on buses would only tally total information for a bus for an entire day, sometimes disaggregated by fare category. This was important for revenue control and ridership reporting, but of little use for planning or other uses. However, there has been dramatic enhancements to the data collected by more recent AFC systems, especially those using smart cards. Data is time-stamped, and increasingly geo-coded. The data can then be matched in post processing to stops. Alternatively, AFC systems are increasingly being specified to include an interface between the AVL and AFC systems so that each AFC event is automatically assigned to the current bus stop identified by the AVL system. As AVL uses an ordered list of stops scheduled to be observed by the bus, it allows “matching” of the AFC event to the correct stop as the AFC data is captured. In addition, the movement of individual smart cards can be tracked through the system, providing a wealth of information on customer behavior, including the possibility of building complete origin-destination matrices. Researchers are also using AFC data as a method for analyzing travel times and system performance.

1.3 CONTEXT DATA

In addition to the above, there is also need to consider data that provides context to the Transit ITS data for purposes of analysis.
Examples include:

• *Major seasonal periods (e.g. university summer break, etc.)*

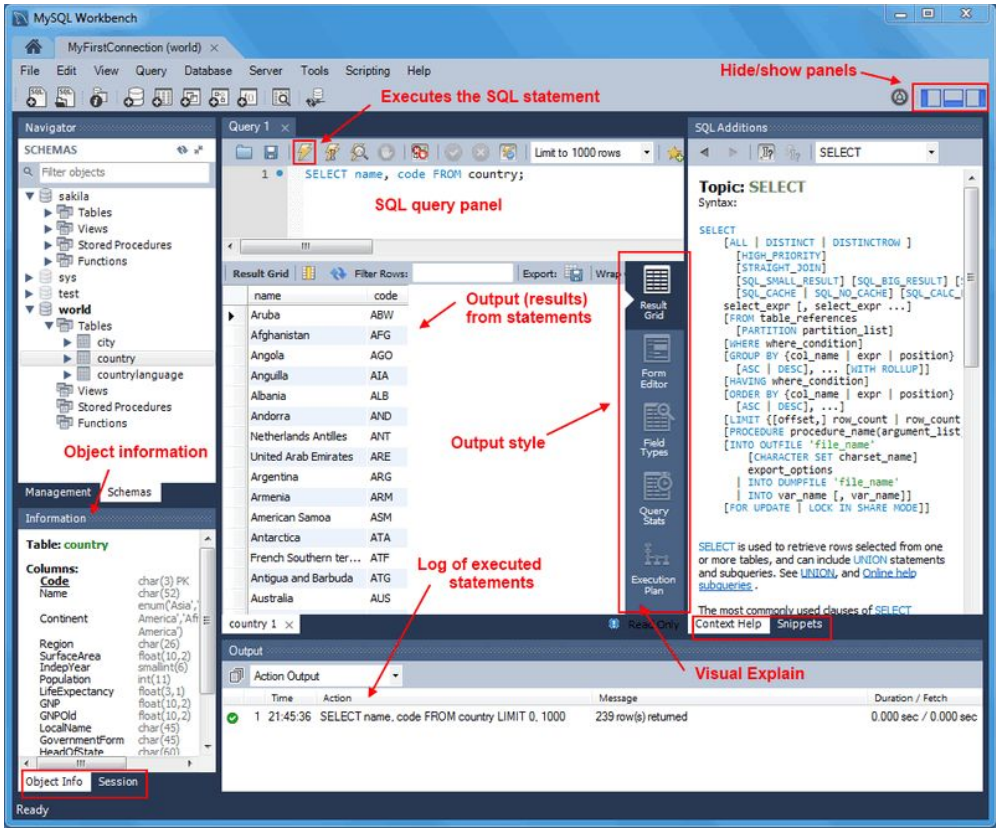
• *Significant weather events*

• *Major events affecting ridership or traffic conditions (e.g. festivals, sports events, natural disasters)*

1.4 Data Entry Stage Processes

Data of ETM is stored in CISTUP server which is divided into different tables according to the Bus depo. The server have data from Dec 2017 to April 2018. But the timeline where every depo operation data is available for Dec 2017 and Jan 2018. That is the reason why we opted for that particular two months in the model we build.

We have used SQL workbench as the interface to fetch the data. Below are the steps to fetch the data using SQL workbench.

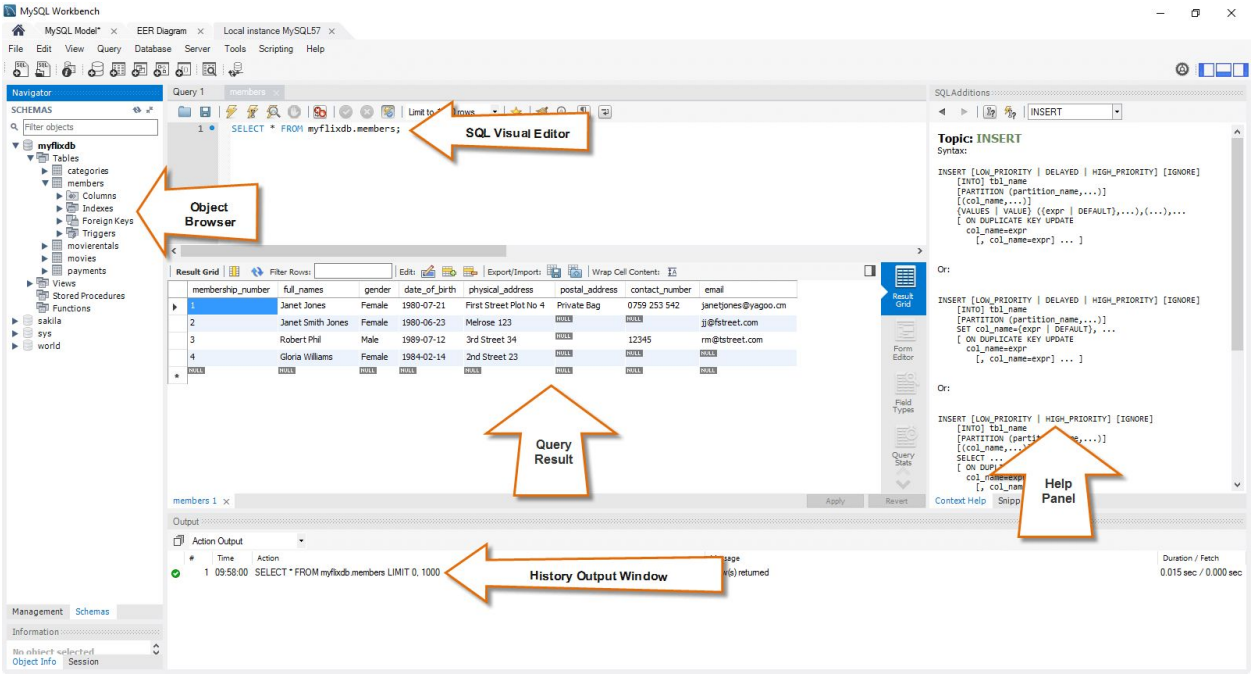


Note : The data shown is not shown as in server due to privacy of BMTC.

- MySQL is an open source relational database that is cross platform.
- MySQL supports multiple storage engines which greatly improve the server performance tuning and flexibility. Prior to version 5.5, the default storage engine was MyISAM which lacked support for transactions, as of version 5.5; the default storage engine is InnoDB which supports transactions and foreign keys.
- MySQL server can be administered using a number of server access mysql tools which include both commercial and open source products. Popular examples include;

    - phpMyAdmin - cross platform web based open source server access tool
    - SQLYog - targeted at the windows platform, desktop commercial server access tool
    - MySQL workbench - cross platform open source server access tool.

- MySQL workbench is an integrated development environment for MySQL server. It has utilities for database modeling and designing, SQL development and server administration.



In the same way we fetch the data from Dec 2017 to Jan 2018 of all the variables required.

Extracting the features- FREQUENCY & WORKING HOURS

```
SELECT
ticket_date,hour(ticket_time),ticket_from_stop_id,COUNT(vehicle_no),vehicle_
no,px_count,route_id,px_total_amount
 FROM ticketd02
 GROUP BY ticket_date,hour(ticket_time),ticket_from_stop_id,vehicle_no
```

Frequency of bus stop - Number of buses arriving to the particular bus stop (SQL)
Working hours of bus stop - Number of hours the bus stop is operating in a day (SQL)

If example the frequency of buses from Bangalore to Mysore.

For travelling From Mysore to Bangalore there are Four types of services are available. With a little break between 12.00am to 1.00am (midnight) every 30 to 45 minutes buses are available to Bangalore-Satellite bus stop. Few buses going towards Kolar,Chikkaballapur, Tamil Nadu side may go to Central stand.

Presently Four types of services are available:

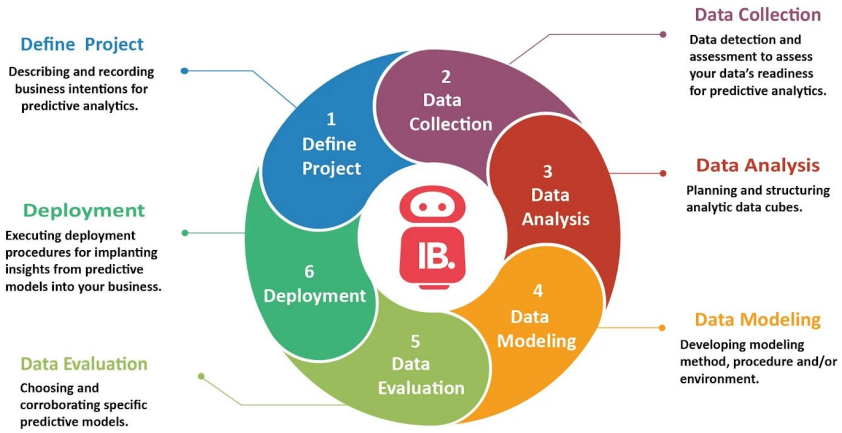
1. VOLVO AC service- Non-Stop : Once in 1 to 1.30 hrs
2. Luxury or Suvihari-Non-Stop: as above-1 service every one hour on weekends
3. Express Non-Stop service: every 30–45 minutes
4. Express Services-every 10 minutes - connecting Bangalore- will have 5 stops and takes 3.5hrs to reach Bangalore
5. Apart from the above Interstate services from Kerala and Tamil Nadu are also available.

The feature “Frequency of bus stop” is calculated on overall basis for the two months and extracted the average out of it.

In the same way the working hours of bus stops resembles that particular bus stop might not get buses for 24/7 timeline, so the number of hours in a day that bus stop is arriving with buses.

1.5 Project Organization

Ridership prediction models show the effect on ridership of various measures such as changes in routes, service frequency, schedules, bus fares, connectivity to and from other modes of travel, improvements in bus stop amenities, and provision of information to travellers. With a daily ridership close to 30 lakhs, BMTC manages a fleet of over 6500 buses. Its buses equipped with GPS and ETM produce large volumes of data on its ridership and route patterns that may be analyzed to provide useful information for the transit company. Similarly, KSRTC buses operating in Mysore have similar ITS features and generate large quantities of data. Bus stops in Mysore are also equipped with electronic boards that show real-time information on its buses. While its ridership is increasing overall, it is losing mode share. Initial analysis of the data from these transit companies was undertaken to observe variations in patterns over weekdays, weekends, public holidays, and during peak and off-peak hours of the day. Overall ridership variations are observed as well as those over a route and in a bus station. Currently schedules of buses are being analysed using the GPS data to understand the effect of reliability on ridership and to identify complementarity and competition between routes. Finally, direct demand models will be estimated for the ridership prediction tool.



Objective of this research is to develop a transit ridership prediction/forecasting tool that can be used to predict changes in transit ridership

Ridership = f (Service characteristics, Time, Location, Pricing ,Other characteristics).

The BMTC role will change

As new mobility options — from micro-mobility to shared rides — enter the public

transit sphere, and since each option has an impact on the economic viability of transit, congestion, and even city-wide economies, transit agencies won't just deal with traditional mass transit (buses) but with regulating and managing multiple modes of transportation and how they affect cities.

Ridership is complex and is influenced by many factors. In 2019 new technologies utilizing big data will help improve transit networks by making them more efficient while reducing the costs. Deriving insights from big data utilizing AI will improve on-time performance and the entire rider experience, from better on-time performance to changing routes, new service offerings, etc. These efforts can and will grow ridership, and cities that will invest in these initiatives will see the numbers turn around.

This is forcing agencies and operators to reconsider what can be done about ridership, forcing a hard look at what determines public transit ridership, from on-time performance, to trip volumes and routes.

Technological advancements and billions of dollars are pouring into a space that for a long while was almost forgotten. This space will keep evolving in the next few years at a pace that will be determined by regulators and their willingness to invest, operators and agencies and their willingness to adapt to market changes and people whose ingenuity and innovation will keep reshaping our lives.

Example as PUNE

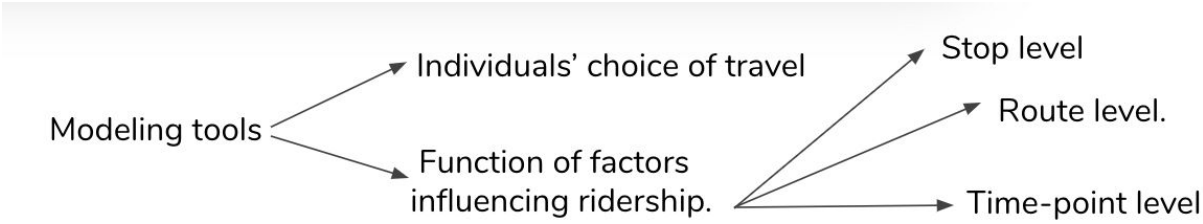
According to the researchers at Parisar, there is a need to decongest roads by increasing the ridership of PMPML buses. "While Pune's population is 3.5 million, the number of registered vehicles has reached 3.62 million. Shifting from cars to public transport will help reduce emissions by 65 per cent during peak hours and 95 per cent during off-peak hours," said Sharmila Deo, who researched on air pollution.

PMPML ridership for 2018 was 10,02,000. Taking it as the base figure, an increase by 25 per cent will make it 12,52,500. Thus, the ridership will double by the end of three years. The transport utility should calculate the buses needed to match the ridership and ensure that it runs on full capacity.

Chapter 2

Literature Review

For the past two decades, intensive research has been and is being conducted in the field of transportation.



The transit ridership modeling can be done in three ways as above flow graph.

Individuals choice of travel - This type of modelling might take huge amount of time because its need to collected as primary data. Sampling might be face a lot of issues while framing the data. This type of modelling comes from customer perspective.

Function of factors influencing ridership - This kind of modelling can be done in three ways and can be done with data from bus service perspective.

Stop level - This kind of modelling is done on the passengers boarded/ alighted per bus stop level. The data framed with ETM can be done on this kind of modelling.

Route level - This kind of modelling needs the data on route level such as traffic density on that particular route. Due to unavailability of data this modelling is not possible in our scenario.

Time- point level - This modelling can be done with the time associated with passenger travelled. This kind of modelling can be associated with stop level.

We have decided to approach the ridership modelling through stop levels in BMTC

(Chu 2004, Chu 2007, Chakour & Eluru, 2016) These are papers in major which took for literature review

Several studies examine transit ridership in an attempt to link ridership with socioeconomic characteristics, built environment, and transit attributes across different contexts. Earlier research has focused on understanding the different factors that affect transit ridership at a macro-level (region or country). Taylor et al. (2009), for example, have undertaken a country-wide study for 265 U.S.

Urbanized areas and concluded that transit ridership is influenced by the regional geography, the metropolitan economy, population characteristics, and the auto/highway system characteristics. The authors have classified the factors that affect transit ridership as internal (fare, level of service) or external (income, parking policies, development, employment, fuel prices, car ownership, and density levels) variables. They observed that external factors generally have a greater effect on ridership than internal factors.

A stream of research examined the effect of trip costs, such as fares, fuel price, and parking price. The elasticity of transit ridership with respect to the fare is negative and inelastic for all transit, and even more so for bus ridership compared to other public transportation modes (Hickey, 2005; Wang and Skinner, 1984). There is also a general consensus that the elasticity of transit ridership with respect to gasoline price is positive and inelastic, especially in medium sized cities (Mattson, 2008; Currie and Phung, 2007).

The price of parking also affects transit ridership; imposing a daily parking fee for commuters will significantly increase transit patronage (Hess, 2001). A set of studies have examined the influence of high gasoline prices between 2005 and 2008 in the United States on transit ridership (for example see Chen et al., 2011; Lane, 2010; Lane, 2012).

Modeling and forecasting transit patronage has recently been advanced to the segment-level (Peng et al. 1997; Kimpel et al. 2000). Segments may be defined by time-point stops as by Kimpel et al. (2000) or by fare zones as by Peng et al. (1997). This advance recognizes the spatial variation of patronage and service supply across the segments of a route. It has been aided by geographic information systems (Peng and Dueker 1995) as well as by the availability of patronage data at the segment level from automated passenger counters.

It has allowed the assessment of new policy instruments such as service reliability (Kimpel et al. 2000). It has also gained new insights into inter-relationships in a transit network and their effects on patronage that traditional route-level analyses were unable

to provide (Peng et al. 1997).

Incorporating the stop level boardings and alightings along various time periods provides us with unique challenges of its own. For instance, the consideration of four time periods for boardings and alightings result in eight dependent variables for each stop. It is important not only to consider different time periods in the analysis, but to assess the possible unobserved interactions between them as well. The dependent variables are all reported for the same stop and hence are likely to be affected by common unobserved factors.

Earlier research efforts on transit ridership estimated a single model for all the transit stops in the urban region. It is possible that there are stops with very high levels of ridership (in the central business district region) and stops with very low levels of ridership (in suburban residential neighborhoods). Considering all stops to be homogenous across the urban region might lead to potential bias in model estimates. Hence, it is useful to identify various categories of stops for an urban region prior to developing statistical models. To be sure, categorizing stops is a city specific process depending on the urban region and transit service in place.

Chapter 3

Methodology and theory

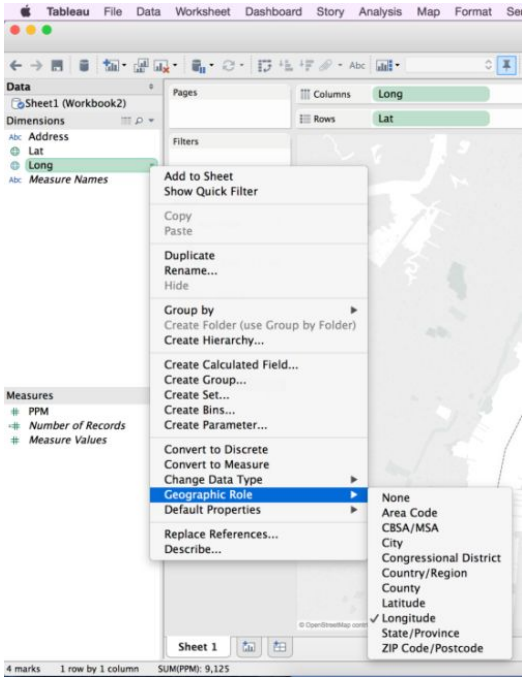
One request that we have when building maps for Tableau is how to quickly generate directions to the points identified in the map, so that our clients can get directions to our listings, stores, etc. The version used in this Tableau map tutorial is Tableau 8.2 but the steps are relevant to other versions as well.

The first step is to get the latitude and longitude of your locations and import them into Tableau. Map will show four real estate listings:

	A	B	C	D	E
	lat	long	address	Rental Amount	
	40.770175	-73.957759	290 East 74th St New York NY 10021	2200	
	40.772179	-73.962551	100 East 74th St New York NY	2300	
	40.771128	-73.95612	300 East 76th St New York NY	2125	
	40.767522	-73.955774	400 East 72nd St New York NY	2500	

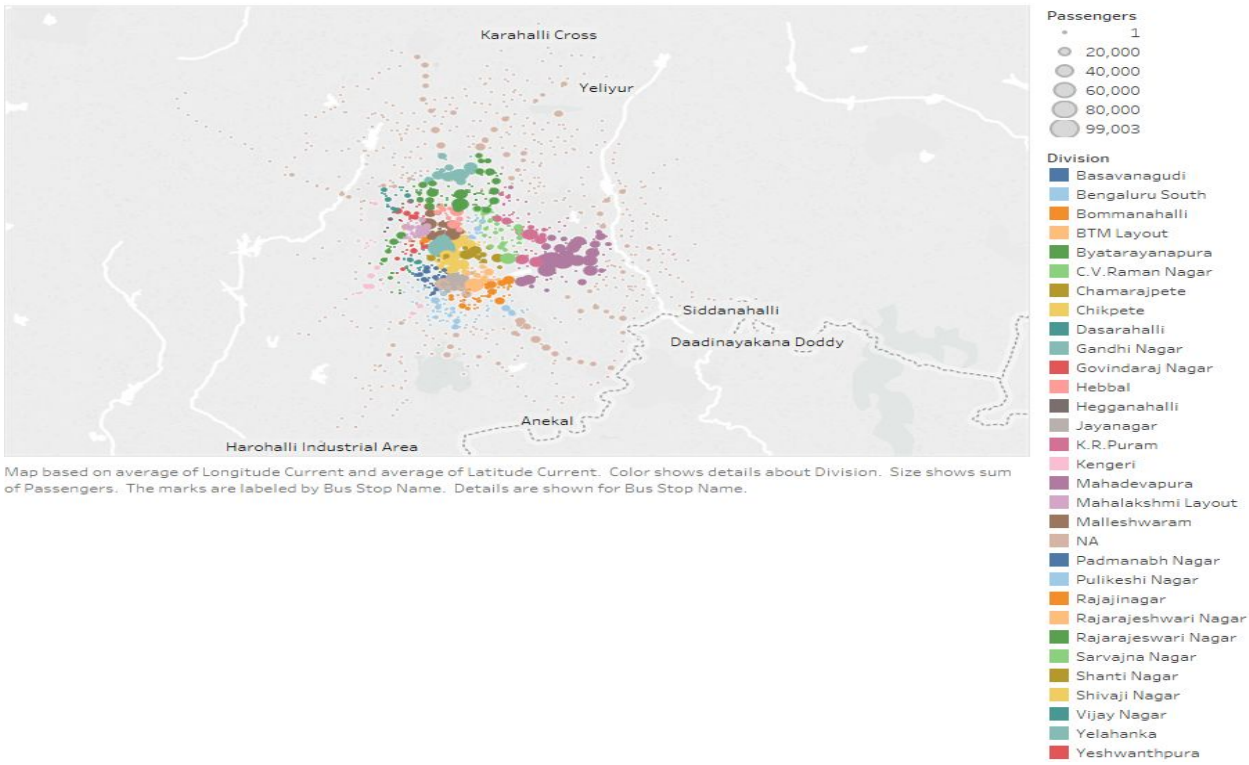
NOTE: Make sure that your latitude and longitude values are in a number format, so that Tableau can read them as geo-coordinates.

After we import the data, we can simply right click on the Latitude and Longitude fields, and from the context menu choose Geographic Role >> and Latitude or Longitude. After converting both your Latitude and Longitude to their respective Geographic Roles,



simply drag the Longitude into the Columns and the Latitude into the Rows shelves and choose the symbol map from the Show Me drawer. In the below map we have also added the Rental Amount to the Color shelf.

Using above steps the ridership of BMTC bus stops is created to know the basic trend before creating the model.



Map based on average of Longitude Current and average of Latitude Current. Color shows details about Division. Size shows sum of Passengers. The marks are labeled by Bus Stop Name. Details are shown for Bus Stop Name.

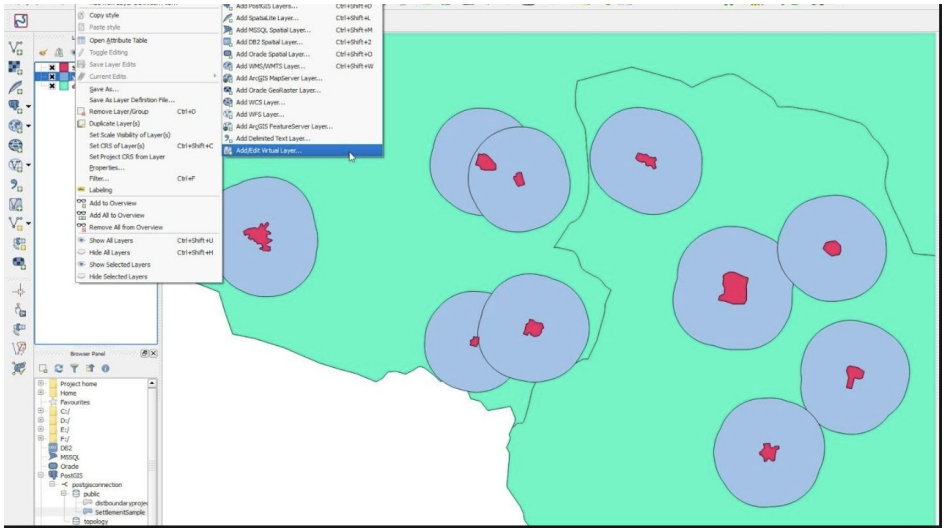
From the above graph we can say that location could be the major factor for ridership at bus stops. So knowing the location characteristics of bus stops would be an important factor for the model we create.

So we use Open street map & QGIS to know the characteristics of the bus stop.

Buffer with QGIS

Exercise using QGIS

This exercise uses the above method as it applies directly in QGIS. Other projects will be welcome to add their own example exercises as well.



1. Launch QGIS

2. Load the data layer. Using menus at top of screen select **LAYER -> Add Vector Layer**
    1. Press **Browse** button
    2. Navigate to the folder holding the sample datasets and select **busstopsall.shp** and press **Open** button
    3. QGIS will automatically zoom to the extent of the bus stop features

3. Run the buffer process from the menu: **TOOLS -> Geoprocessing Tools -> Buffer(s)**
    1. Set the **Input vector layer** to **busstopsall**
    2. Enter **Buffer distance** of **300**. This will be 300 metres.
    3. Set the **Output shapefile** by pressing **Browse** button. Navigate to the data folder and enter the new filename as **busstops_300m_buffer.shp**. Press **Save** button to select that filename.
    4. Press **OK** button to run the process.

4. After running, the system asks **Would you like to add a new layer to the TOC?** select **Yes** and the resulting shapefile will be added as a layer in your map view.

5. Buffer(s) window remains open. Press **Close** to close the window.

6. Reorder the layers so the resulting buffer polygons layer is on the bottom of the layer stack. Click and drag the *busstops_200m_buffer.shp* layer to beneath *busstopsall*.

In the above mentioned steps we create a buffer around bus stops using QGIS and keep it aside. Now we use Open street map to know the number of hospitals,schools etc whatever we need to sync up with the layer we created in bus stop.

A special phrase just means that nominatim will give some preference to items tagged amenity=hospital, it's still doing name matching on "Hospitals", so it's returning things with exactly that name.

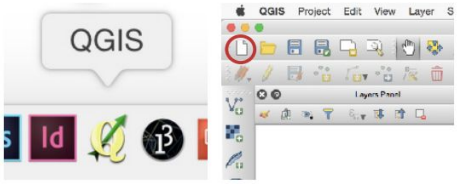
Overpass-API query that returns hospitals within that location. Note that you can use that query outside of Overpass Turbo, it's just a convenient way to share it.

You could also compute a bounding box from the coordinate and restrict the search to that area, if that were more suitable for your goal.

Exporting map data from OpenStreetMap

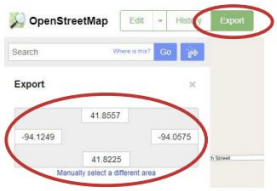
This task sheet presents an open source workflow that uses crowd-sourced data from OpenStreetMap (OSM) to create GeoJSON files that can be used on web mapping platforms like leaflet or mapbox. Using free QGIS software, users can download data from OSM, and then filter and edit it within the QGIS environment. The data can be used for analysis, in map layouts, or exported to various file formats, including GeoJSON.

1. Downloading OpenStreetMap Data

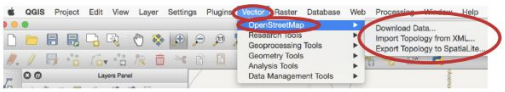


a. In order to download OSM data, you will need to define the extent of the area for which you wish to download data. You can do this in three ways: from the map canvas, from a layer, or manually by adding the coordinates. We will define the extent manually.

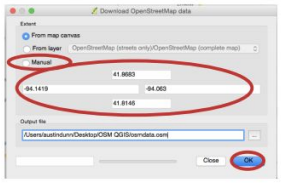
b. Go to OpenStreetMap.org and navigate the web map to your area of interest. Click on the **Export** button to see the coordinates of your current map extent. *Note: the larger the extent, the more time it will take to download and process the data.*



c. Open QGIS on your PC or Mac and proceed to open a new project. Click on **Vector** > **OpenStreetMap** > **Download Data**. In the **Download OpenStreetMap Data** window, click **Manual** and manually enter the extent identified in **step 1b**. Save the output .osm file in an appropriate location and click **OK** and close the window after the download is successful.

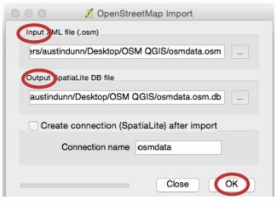


2. Importing Topology



a. Click **Vector** > **OpenStreetMap** > **Import Topology from XML**.

b. In the **OpenStreetMap Import** window under **input XML file** select the .osm file created in **step 1c**. Choose a name and location for the **Output SpatialLite DB file** and click **OK**. Close the window after the import is successful.



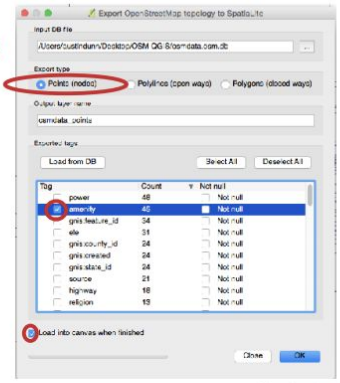
3. Exporting OSM Topology to SpatialLite

a. Click **Vector** > **OpenStreetMap** > **Export Topology to SpatialLite**.

b. In the **Export OpenStreetMap topology to SpatialLit** window select the file created in **step 2b** as the **Input DB file**.

c. Each geometry type: point (nodes), polylines (open ways), and polygons (closed ways) must be exported to the database individually and you can choose what type of features to add based on the OSM **tags**. To learn about OSM tags, visit wiki.openstreetmap.org/wiki/Tags.

d. First, select **Points (nodes)**, and click **Load from DB** to see the tags. Select the **amenity** tag to export features that are tagged as amenities. Select **Load into canvas when finished** and click **OK** and close the window. *Note: you can do this again for each of the geometry types and select as many tags as you are interested in.*

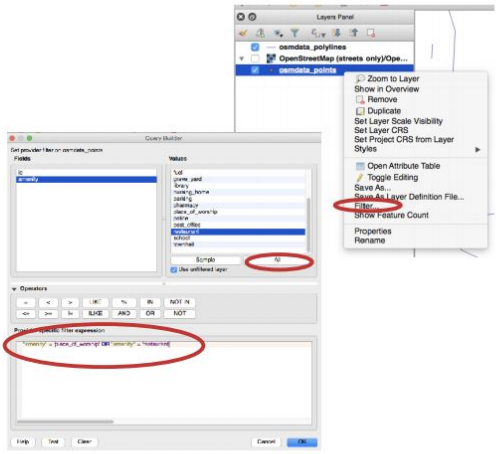


4. Filter the Data

a. To further filter the newly added data, **right-click** on the layer from the **Layers Panel** and select **Filter**.

b. In the **Query Builder** window double-click **amenity** from the list of fields and click **All** to load in all the values. Double-click on a field or value to add it to the **filter expression** box. Use the fields, values, and operators to create an expression that reads: **`"amenity" = 'place of worship' OR "amenity = 'restaurant'`**
Note: depending on your area of interest, you may not have these values available from the values field.

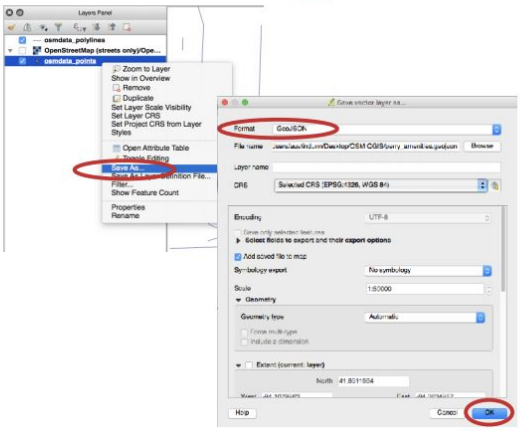
c. Click **OK**. Now the map will only display points for restaurants and places of worship.



5. Export the data as a GeoJSON

a. To export the layer as a GeoJSON file, right-click on the layer from the **Layer Panel** and select **Save As**.

b. Select **GeoJSON** as the **Format**, and create a name and designated location for the file. Click **OK**. *Note: there are over 20 different file formats to choose from.*



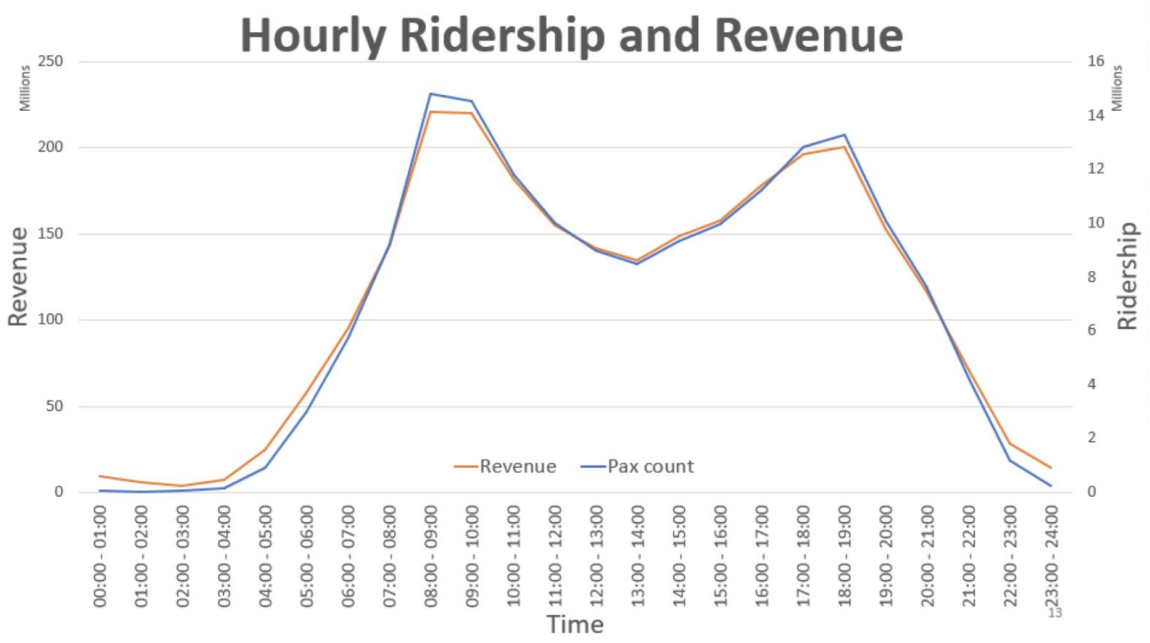
In the same way we need to extract the features of hospitals,residents,schools etc around the bus stops with the buffer of 300 meters. And we should full join the data of BMTC bus stops and features extracted using QGIS & openstreetmap. Then we can create the model on the dataset created regarding the bus stops. The dataset will be attached in the report.

Chapter 4

Proposed Process

Boarding modelling within bus stop

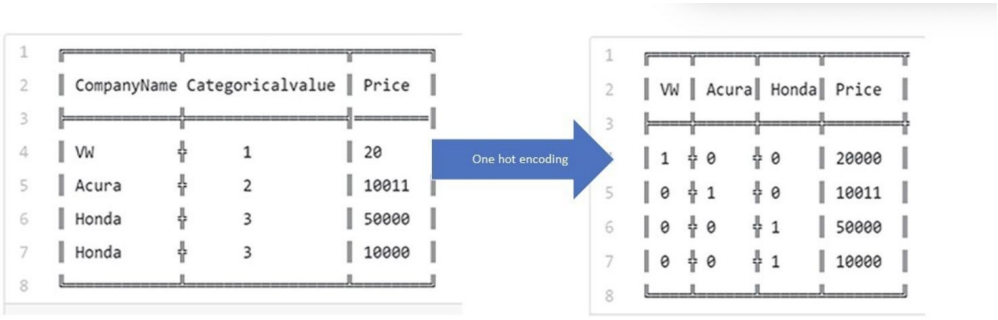
To predict the passengers of a particular bus stop, we take single bus stop data with the factors of whether it is a weekday or weekend & peak hour or non peak hour. Types of peak is taken on the average distribution of ridership in bangalore.



The rise and downfall of curve is called “Off peak”, the high pitch is known as “Heavy peak” and remaining flat curve is known as “Non Peak”.

We do the above factors as one hard coding as those are categorical variables.

One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.



One hot encoding explained in an image

Some algorithms can work with categorical data directly.

For example, a decision tree can be learned directly from categorical data with no data transform required (this depends on the specific implementation).

Many machine learning algorithms cannot operate on label data directly. They require all input variables and output variables to be numeric.

In general, this is mostly a constraint of the efficient implementation of machine learning algorithms rather than hard limitations on the algorithms themselves.

This means that categorical data must be converted to a numerical form. If the categorical variable is an output variable, you may also want to convert predictions by the model back into a categorical form in order to present them or use them in some applications.

As a predictive analysis, the multiple linear regression is used to explain the relationship between oRidership and day/time independent variables. The independent variables can be continuous or categorical. But in our case we encoded as a 0 and 1.

Multiple coefficients to determine and complex computation due to the added variables.

$$Yi=\alpha+\beta 1xi(1)+\beta 2xi(2)+....+\beta nxi(n)$$

Yi is the estimate of ith component of dependent variable y, where we have n independent variables and xij denotes the Ith component of the jth independent variable/feature. Similarly cost function is as follows,

$$E(\alpha,\beta 1,\beta 2,...,\beta n)=12m\sum i=1m(yi-Yi)$$

where we have m data points in the training data and y is the observed data of dependent variable.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              ridership   R-squared:                       0.618
Model:                            OLS   Adj. R-squared:                  0.610
Method:                 Least Squares   F-statistic:                     73.45
Date:                Sun, 16 Jun 2019   Prob (F-statistic):           2.65e-28
Time:                        15:13:59   Log-Likelihood:                -645.61
No. Observations:                 140   AIC:                             1299.
Df Residuals:                     136   BIC:                             1311.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         19.0239      2.029      9.375      0.000      15.011      23.037
Weekday        8.7807      2.347      3.741      0.000       4.139      13.423
Weekend       10.2432      3.359      3.049      0.003       3.600      16.886
Heavy peak    56.9302      3.040     18.727      0.000      50.918      62.942
Non peak     -21.5546      5.576     -3.866      0.000     -32.581     -10.528
Off peak     -16.3518      4.705     -3.475      0.001     -25.657      -7.047
==============================================================================
Omnibus:                       40.812   Durbin-Watson:                   1.607
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               84.812
Skew:                          -1.258   Prob(JB):                     3.83e-19
Kurtosis:                       5.864   Cond. No.                     2.80e+16
==============================================================================
```

The p-value for each term tests the null hypothesis that the coefficient is equal to zero (no effect). A low p-value (< 0.05) indicates that you can reject the null hypothesis. In other words, a predictor that has a low p-value is likely to be a meaningful addition to your model because changes in the predictor's value are related to changes in the response variable.

Conversely, a larger (insignificant) p-value suggests that changes in the predictor are not associated with changes in the response.

In the output above, we can see that the predictor variables of all are significant because of their p-values are almost 0.000. However, the prediction of single bus stop can be modelled on these parameters and the same model is applied on different bus stops. Then the R squared value is almost the same of others.

```
Mean Absolute Error: 17.907227813253265
Mean Squared Error: 606.3131002659293
Root Mean Squared Error: 24.623425843410363
```

Different machine learning algorithms are applied on the same data on different bus stops. There is a significant increase in R squared value comparing with linear regression.

The algorithm of Random Forest

Random forest is like bootstrapping algorithm with Decision tree (CART) model. Say, we have 1000 observation in the complete population with 10 variables. Random forest tries to build multiple CART models with different samples and different initial variables. For instance, it will take a random sample of 100 observation and 5 randomly chosen initial variables to build a CART model. It will repeat the process (say) 10 times and then make a final prediction on each observation. Final prediction is a function of each prediction. This final prediction can simply be the mean of each prediction.

```
Out[321]: 0.6731247863803782
```

The model is able to separate the train and validation sets with a r-square value 0.67

The algorithm of Gradient boosting

Instead of parameters, we have weak learner sub-models or more specifically decision trees. After calculating the loss, to perform the gradient descent procedure, we must add a tree to the model that reduces the loss (i.e. follow the gradient). We do this by parameterizing the tree, then modifying the parameters of the tree and moving in the right direction by (reducing the residual loss).

```
Out[324]: 0.6732025954469272
```

Parameters / levers to tune Random Forests

Parameters in random forest are either to increase the predictive power of the model or to make it easier to train the model.

Machine learning tools like random forest, SVM, neural networks etc. are all used for high performance. They do give high performance, but users generally don't understand

how they actually work. Not knowing the statistical details of the model is not a concern however not knowing how the model can be tuned well to clone the training data restricts the user to use the algorithm to its full potential.

```
Model Performance
Average Error: 9.4624 degrees.
Accuracy = 46.67%.
Improvement of 1.64%.
```

Boarding modelling between bus stops

But predicting ridership for single bus stop may not be so sourceful. We need to predict the ridership for array of bus stops. So that we need the characteristics of bus stops. We have already mentioned in Data entry stage process that how we created the characteristics of bus stops. Below are the characteristics of bus stop marked in the buffer of 200 mts.

Residentials	OPS & QGIS
No. of School	OPS & QGIS
No of Colleges	OPS & QGIS
Industries/Commercials	OPS & QGIS
No. of Hospitals	OPS & QGIS
Metro/ Railway	OPS & QGIS

*OPS - Open street map

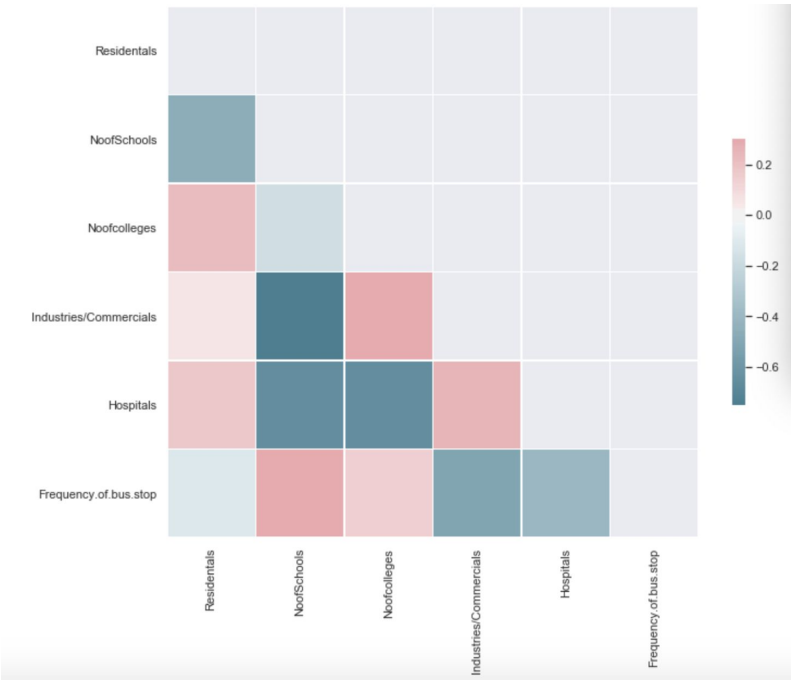
Correlation Analysis

In correlation analysis, we estimate a sample correlation coefficient, more specifically the Pearson Product Moment correlation coefficient. The sample correlation coefficient, denoted r,

ranges between -1 and +1 and quantifies the direction and strength of the linear association between the two variables. The correlation between two variables can be positive (i.e., higher levels of one variable are associated with higher levels of the other) or negative (i.e., higher levels of one variable are associated with lower levels of the other).

The sign of the correlation coefficient indicates the direction of the association. The magnitude of the correlation coefficient indicates the strength of the association.

For example, a correlation of r = 0.9 suggests a strong, positive association between two variables, whereas a correlation of r = -0.2 suggest a weak, negative association. A correlation close to zero suggests no linear association between two continuous variables.



Few bus stops with interest is considered and sent as array to a data frame and different machine learning models were applied on it.

```
highrangebusstops =df_final[df_final['Bus_stop_id'].isin([47,124,193,268,336,411,980,4311])]
```

In this case regression is not giving as good results as for single bus stop because the data is not linear.

"If our data is strongly non-linear" use non-linear methods to model your working variables relations. I consider that in this case the problem requires numerical methods and fitting tests from mathematics, not from statistics.

If you estimate the distribution of each variable, try to work each one with non parametric methods. In this case I use the Laplace premise that assigns frequence 1/N to each measured Xi value, and build Lorenz curves for each variable. I suppose that your datasets are "representative" enough; that means that the sample gives X values not to far from averages of each ordered value with interval frequence 1/N, which produce a good U estimated mean, and acceptable shapes of distribution curves.

Practically, in almost all cases, if you have to choose one method. Boosted Trees (GBM) is usually be preferred than RF if you tune the parameter carefully.

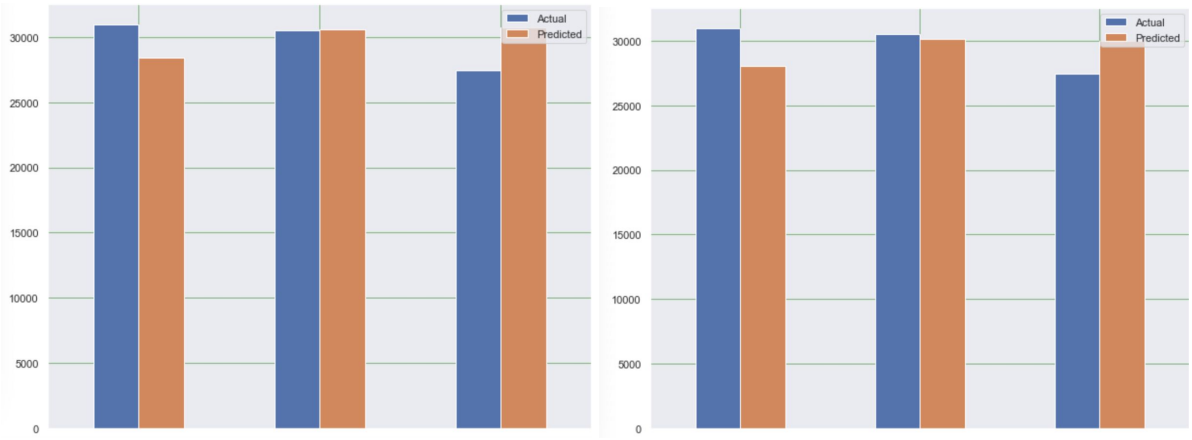
The major reason is in terms of training objective, Boosted Trees(GBM) tries to add new trees that compliments the already built ones. This normally gives you better accuracy with less trees.This being said, the ideas of subsampling and bagging in RF is important. They can readily be incorporated into boosted tree training. This will indeed help the performance usually.

There is a historical reason such that RF is easier to parallelize. This can also be done for boosted trees, though less trivial. Boosted Trees can be distributed and very fast. We did it in dmlc/xgboost and it works pretty well.

Random Forest

Gradient Boosting

Bus stop id- 47 193 980 47 193 980



One last advantage of boosted trees are about modeling, because boosted trees are derived by optimizing an objective function, basically it can be used to solve almost all objective you can write gradient out. This includes things like ranking, poisson regression, which RF is harder to achieve

Feature Importance

The data features that you use to train your machine learning models have a huge influence on the performance you can achieve.

Irrelevant or partially relevant features can negatively impact model performance.

Bagged decision trees like Random Forest and Extra Trees can be used to estimate the importance of features.

In the example below we construct a ExtraTreesClassifier classifier for the characteristics of bus stop.

```
# Feature Importance with Extra Trees Classifier
from pandas import read_csv
from sklearn.ensemble import ExtraTreesClassifier
# feature extraction
model = ExtraTreesClassifier()
model.fit(X, Y)
print(model.feature_importances_)
```

```
[0.3   0.086 0.071 0.214 0.086 0.243 0.   ]
```

'Residentials':0.3,

'NoofSchools':0.086

,'No Of Colleges':0.071 ,

'Industries/Commercials':0.214 ,

'Hospitals':0.086 ,

'Frequency.of.bus.stop':0.243

'Working.hours':0.

You can see that we are given an importance score for each attribute where the larger score the more important the attribute. The scores suggest that the importance of Residentials,Industries/Commercials and Frequency of bus stop.

If we apply only the above features in the model we might get the best results because

Three benefits of performing feature selection before modeling your data are :

- Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.
- Improves Accuracy: Less misleading data means modeling accuracy improves.
- Reduces Training Time: Less data means that algorithms train faster.

The linear regression model may not likely to be appropriate for this research effort. The distributions of stop boarding, alighting, and total activities are extremely skewed toward the origin.

Boarding in one direction in the morning peak is usually correlated with alighting in the opposite direction in the evening peak. This needs to be addressed if both boardings and alightings are modeled.

It is important because there are so many prediction problems that involve a time component. These problems are neglected because it is this time component that makes time series problems more difficult to handle.

Predictions are made for new data when the actual outcome may not be known until some future date. The future is being predicted, but all prior observations are almost always treated equally. Perhaps with some very minor temporal dynamics to overcome the idea of “concept drift” such as only using the last year of observations rather than all data available.

Time does play a role in Transit datasets.A time series dataset is different.Time series adds an explicit order dependence between observations: a time dimension.

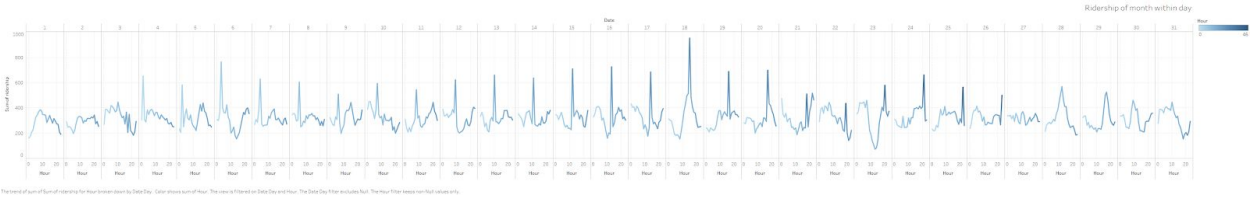
This additional dimension is both a constraint and a structure that provides a source of additional information.

In descriptive modeling, or time series analysis, a time series is modeled to determine its components in terms of seasonal patterns, trends, relation to external factors, and the like.….In contrast, time series forecasting uses the information in a time series (perhaps with additional information) to forecast future values of that series

The purpose of time series analysis is generally twofold: to understand or model the stochastic mechanisms that gives rise to an observed series and to predict or forecast the future values of a series based on the history of that series

The skill of a time series forecasting model is determined by its performance at predicting the future. This is often at the expense of being able to explain why a specific prediction was made, confidence intervals and even better understanding the underlying causes behind the problem.

The below plot is ridership of BMTC in the month of december per 24 hour level. We can see the day 1 ridership of that month in the first block.



1) Decompose your data

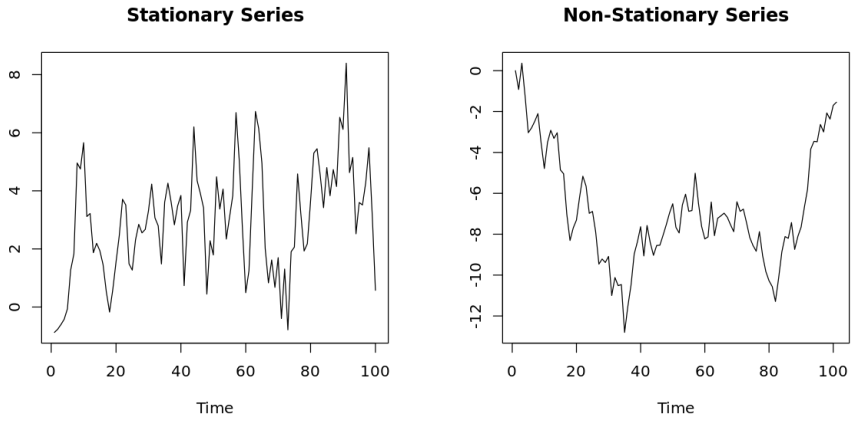
   Does the series appear to have trends or seasonality?

   Use decompose() or stl() to examine and possibly remove components of the series

2) Stationarity

   Is the series stationary?

Fitting an ARIMA model requires the series to be stationary. A series is said to be stationary when its mean, variance, and autocovariance are time invariant. This assumption makes intuitive sense: Since ARIMA uses previous lags of series to model its behavior, modeling stable series with consistent properties involves less uncertainty. The left panel below shows an example of a stationary series, where data values oscillate with a steady variance around the mean of 1. The panel on the right shows a non-stationary series; mean of this series will differ across different time windows.



The augmented Dickey-Fuller (ADF) test is a formal statistical test for stationarity. The null hypothesis assumes that the series is non-stationary. ADF procedure tests whether the change in Y can be explained by lagged value and a linear trend. If contribution of the lagged value to the change in Y is non-significant and there is a presence of a trend component, the series is non-stationary and null hypothesis will not be rejected.

Use adf.test(), ACF, PACF plots to determine the order of differencing needed

```
> adf.test(count_ma, alternative = "stationary")

        Augmented Dickey-Fuller Test

data:  count_ma
Dickey-Fuller = -10.461, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary
```

3) Autocorrelations and choosing model order

   Choose the order of the ARIMA by examining ACF and PACF plots

4) Fit an ARIMA model

The forecast package allows the user to explicitly specify the order of the model using the arima() function, or automatically generate a set of optimal (p, d, q) using auto.arima(). This function searches through combinations of order parameters and picks the set that optimizes model fit criteria.There exist a number of such criteria for comparing quality of fit across multiple models. Two of the most widely used are Akaike information criteria (AIC) and Baysian information criteria (BIC). These criteria are closely related and can be interpreted as an estimate of how much information would be lost if a given model is chosen. When comparing models, one wants to minimize AIC and BIC.

While auto.arima() can be very useful, it is still important to complete steps 1-5 in order to understand the series and interpret model results. Note that auto.arima() also allows the user to specify maximum order for (p, d, q), which is set to 5 by default.We can specify non-seasonal ARIMA structure and fit the model to deseasonalize data. Parameters (1,1,1) suggested by the automated procedure are in line with our expectations based on the steps above; the model incorporates differencing of degree 1, and uses an autoregressive term of first lag and a moving average model of order 1:

Using the ARIMA notation introduced above, the fitted model can be written as

$$\Delta y_t = 20.5740 + 1.7487 \Delta y_{t-1} - 0.8148 \epsilon_{t-1} + \epsilon_t$$

where E is some error and the original series is differenced with order 1.

AR(1) coefficient p = 20.57 tells us that the next value in the series is taken as a dampened previous value by a factor of 20.57 and depends on previous error lag.

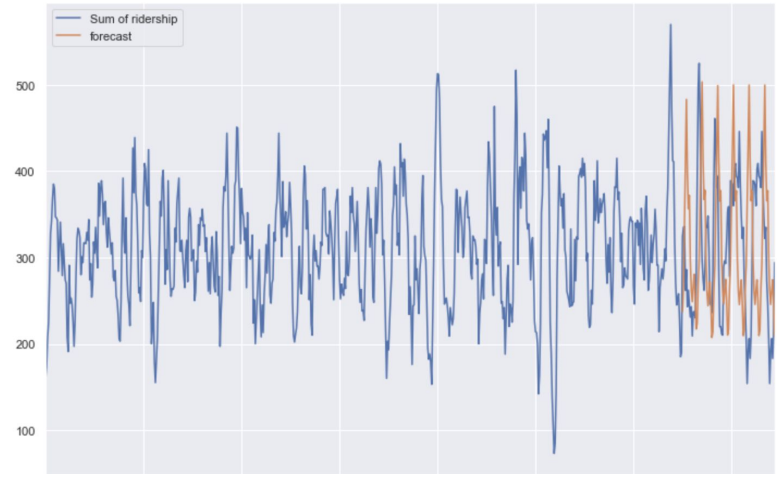
```
> ForecastsArima
 [1] 304.2853 293.2258 294.1817 293.3840 289.7886 300.8076 304.4932 305.6939 325.7375 316.8018 317.8411
[12] 330.5868 313.4730 308.2826 307.0051 317.1975 327.1998 294.2283 299.8538 301.1819 317.3477 313.2428
[23] 323.7359 324.8233
```

Although ARIMA is a very powerful model for forecasting time series data, the data preparation and parameter tuning processes end up being really time consuming. Before implementing ARIMA, you need to make the series stationary, and determine the values of p and q using the plots we discussed above.

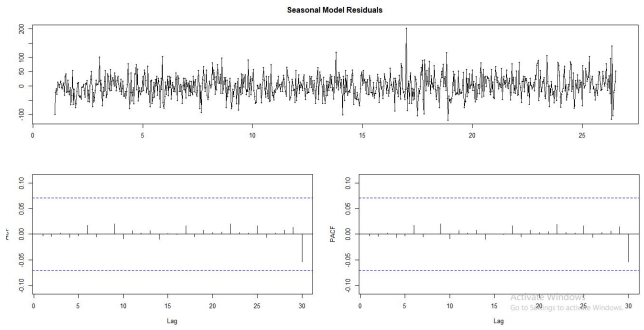
```
> ResultsArima
   p d q P Q     MAPE      AIC
58 2 0 1 0 1 10.20685 7500.119
66 2 0 2 0 1 10.20959 7501.614
67 2 0 2 1 0 10.20959 7501.614
60 2 0 1 1 1 10.20873 7502.079
42 1 0 2 0 1 10.60642 7542.607
44 1 0 2 1 1 10.61958 7544.376
50 2 0 0 0 1 10.67008 7550.862
52 2 0 0 1 1 10.67673 7552.798
68 2 0 2 1 1 10.68851 7553.293
34 1 0 1 0 1 10.71307 7553.811
36 1 0 1 1 1 10.72063 7555.751
26 1 0 0 0 1 10.81410 7558.648
28 1 0 0 1 1 10.82391 7560.573
6  0 1 0 0 1 11.08640 7626.933
8  0 1 0 1 1 11.09729 7628.808
30 1 1 0 0 1 11.08639 7628.915
31 1 1 0 1 0 11.08639 7628.915
32 1 1 0 1 1 11.08639 7628.915
14 0 1 1 0 1 11.08593 7628.916
22 0 1 2 0 1 11.09004 7630.049
54 2 1 0 0 1 11.09139 7630.126
16 0 1 1 1 1 11.10066 7630.791
38 1 1 1 0 1 11.09206 7630.915
39 1 1 1 1 0 11.09206 7630.915
40 1 1 1 1 1 11.09206 7630.915
46 1 1 2 0 1 11.08462 7631.395
47 1 1 2 1 0 11.08462 7631.395
48 1 1 2 1 1 11.08462 7631.395
62 2 1 1 0 1 11.08768 7631.432
63 2 1 1 1 0 11.08768 7631.432
64 2 1 1 1 1 11.08768 7631.432
24 0 1 2 1 1 11.11138 7631.882
70 2 1 2 0 1 11.09368 7633.790
```

Returns best ARIMA model according to either AIC, which is (2,0,1)

Forecasting using a fitted model is straightforward in R. We can specify forecast horizon h periods ahead for predictions to be made, and use the fitted model to generate those predictions:



So now we have fitted a model that can produce a forecast, but does it make sense? Can we trust this model? We can start by examining ACF and PACF plots for model residuals. If model order parameters and structure are correctly specified, we would expect no significant autocorrelations present.



We calculate MAPE on validation data. We run the ARIMA on validation data with all selected P and Q.

Mean Squared Percentage Error (MAPE) for each model :

MAPE = Abs(Actual – Predicted) / Actual *100 = **2.157301**

Chapter 5

Conclusion & Future Scope

After applying various linear and machine learning models on the data available we can consider that Time series models are doing a pretty good job in predicting ridership at various levels. But the data available which is Dec 2017 & Jan 2018 might be the less time period to apply ARIMA models. If we manage to get the data of minimum 6 months we can able to predict the ridership at accurate level. When comes to machine learning models like random forest the bus stop characteristics helps a lot in defining ridership at bus stop levels. If we are able to manage to get the traffic density data at route levels we can predict the ridership according to route levels.

For the future scope of project the complete code and data achieved is uploaded in the below github link.

https://github.com/Revanthgh/BMTC-Ridership-prediction-at-bus-stop-level/upload/master

Data

https://indianinstituteofscience-my.sharepoint.com/personal/guthala_iisc_ac_in/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fguthala_iisc_ac_in%2FDocuments%2FCISTUP