Indiana University Bloomington

CSCI B 565

Data Mining

Data Analytics for IU Bus System

Group Name : RedMiners

Authors:
Jayasankar, Siddharth
Nagarajan, Ganesh
Madhavan, Sarvothaman

Supervisor:
Dr. Dalkilic, Mehmet

December 18, 2015

Abstract

University Bus systems are widely used for shuttling students, faculty and patrons in and out of University campus making it the easiest mode of transportation to move around the University. Indiana University operates its bus system under the name IU Bus[1]

The project that this paper reports on attempts to develop an uniform framework and processes for analyzing IU Bus data and thus along the process hopes to establish a general framework. Any such framework should be able to transform practical nuances into presentable data analytics and also be scalable. The areas focused in this paper consider various aspects of discussion with the key stake holders, developing glossary, developing metrics for measuring performance,data pipe lining, data visualization, Model creation and model verification.

This project uses MySql as its preferred database for storing operational data, Neo4j graphical database for holding aggregated data, Tableau Public for visualizing the data and R for model creation and verification.

Contents

1 Introduction

University Bus systems are critical for shuttling students in and out of university campuses. These bus systems are not only used at Indiana University, but also at other university systems like University Transit Service,University of Virginia[2], The Bus, Boston University system [3] ,etc. All these universities have similar kind of schedules for different semesters like Spring, Fall and special trips during late nights. All these services have an live tracking system which records the bus statuses, Geo-locations of buses and their stops and is presented real-time to the users. We also consider Cintra, Neves[4] as one of the approaches for prediction

IU Bus system has four routes A,B,E and X and has about 52 stops including the minor and major stops. These busses are scheduled about one bus in every 15 minutes for use of its passengers. IU Bus System commits itself to stick to the schedule[5], however as with any transportation system, the bus gains or looses speed during the course of the travel, thereby deviating form the actual schedule.

Thus, primary interest of this study is to understand the time difference between the schedule time and the actual time, $t_{schedule} - t_{actual}$. An model was created in Neo4j depicting the acutal bus system, and the ability of the graph database serves in the shortest paths problem and going from place A to place B.

In addition to the above said studies, an statistical model was developed in R to predict the arrival status of the bus at a given stop. The statuses can be On-Time, Delayed and Early, an detailed explanation is given in the corresponding sections. In addition to these, a reporting solution was developed in Tableau and is hosted on cloud at location given below. Here, detailed reporting of status of each trips for Spring 2015 for Route A is available. This includes the bus arrival status and the seconds by which the bus is early / on-time.

Following are the locations of the project artifacts:

1. The database schema is part of this report and the corresponding SQLs

are available at github url below

2. Neo4j Cloud Database : `http://iubus.sb02.stations.graphenedb.com:24789/browser/`

3. Tableau Public : `https://public.tableau.com/profile/ganesh.nagarajan#!/vizhome/RouteA/IUBus`

4. GitHub Project URL : `https://github.com/sarvothaman/nefarious-octo-rutabaga`

All the codes, data extracts, models, templates related to the project are available at the Github location specified above.

2 Problem Description

The main objectives that were considered are

1. Find out how early/late each trip of each route was in the past.

2. Use the both GPS and schedule data at hand to try improve the performance and perception of the IU transportation system

Our project uses a subset of data available to create a framework which can then be employed to perform a variety of activities. The data used

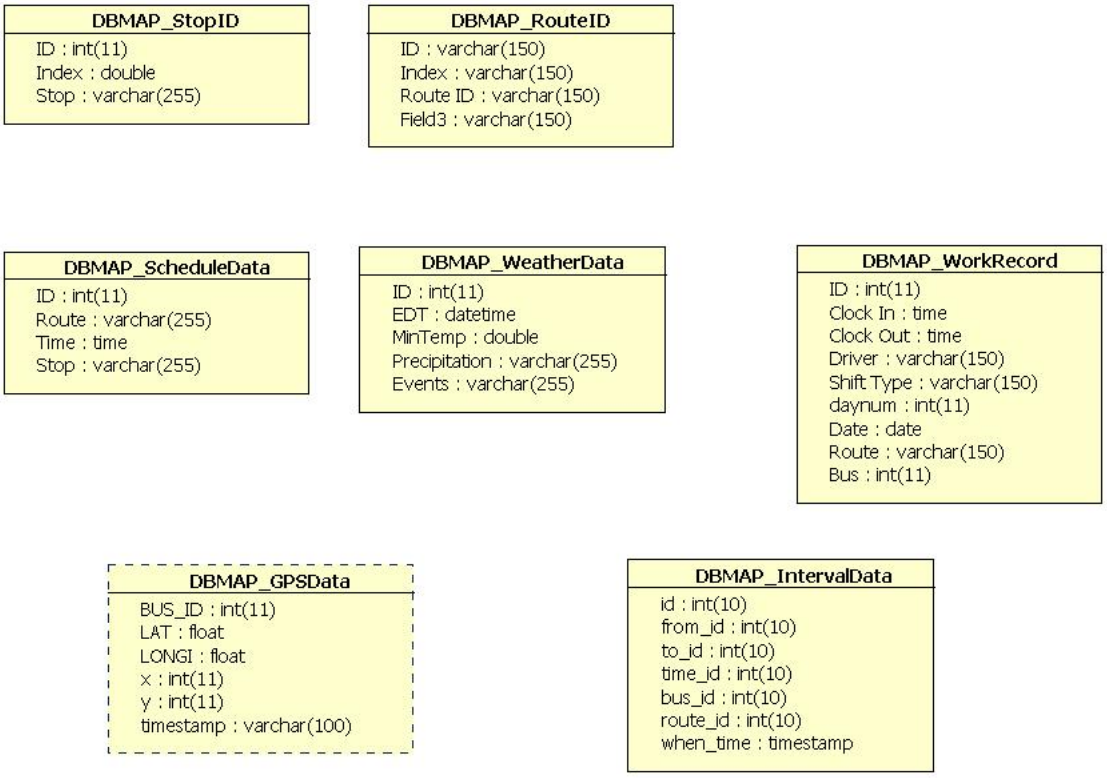
1. is from the spring semester of 2015

2. contains data from Monday through Thursday for all regular routes (does not include nightowls, weekend routes etc)

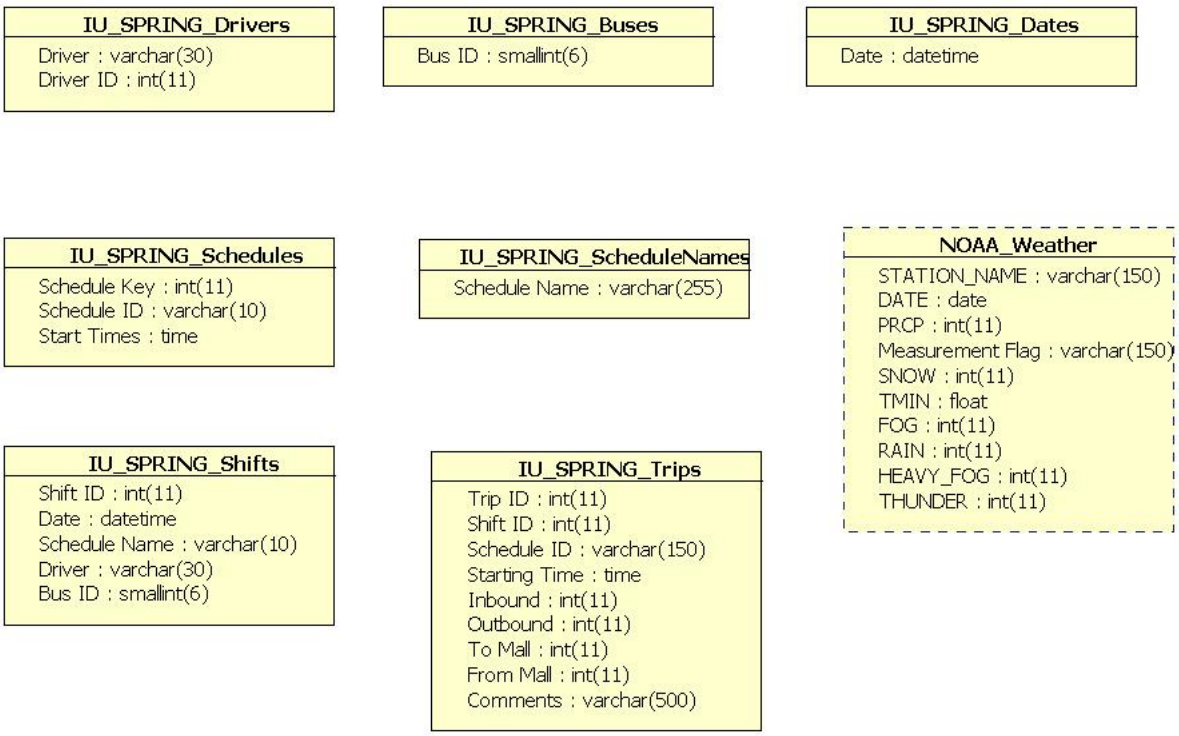
3 Data Management

3.1 Source Data Model

The source data which was made available in Micorosft Access DB was imported into MySQL database to facilitate manipulation. The data model into which the data was imported is shown below:



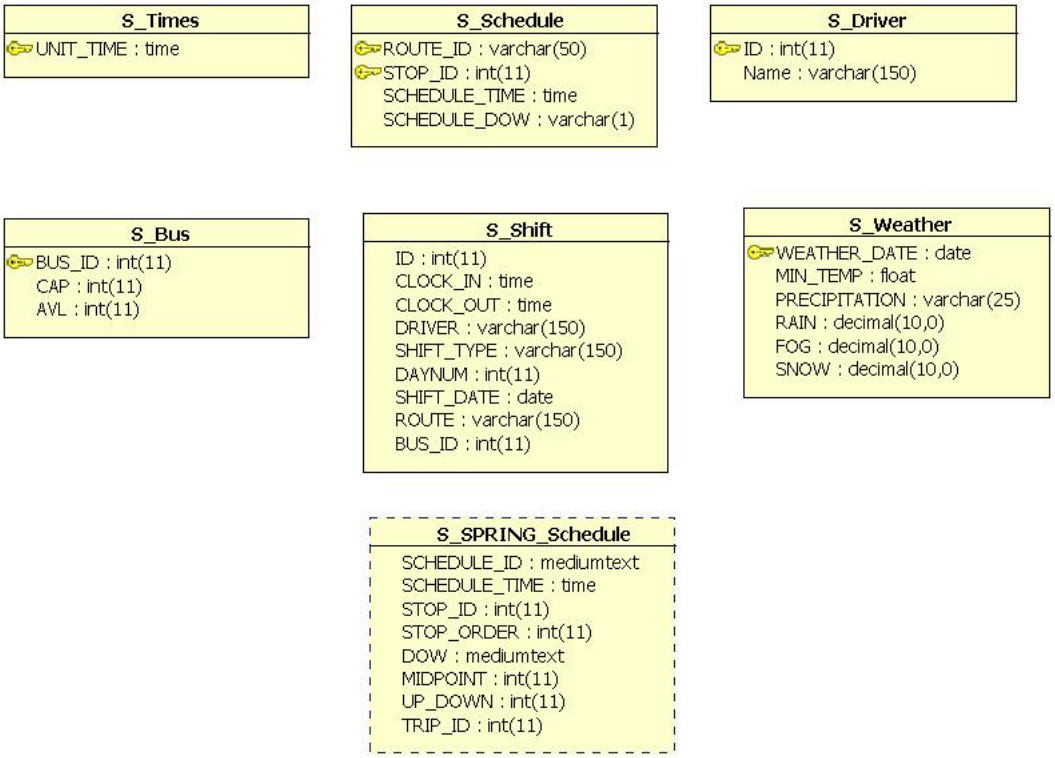
The tables that are prefixed 'DBMAP__'contain data from DoubleMap. This includes but not limited to schedules, work records (shifts and drivers), Time and Location for each trip for each route with bus that was run in that particular route.



The tables that are prefixed 'IU__'contain data maintained by IU Bus System. This included schedules, trips and their overview such as inbound and outbound passenger count. The table prefixed 'NOAA__'contains weather data which was obtained from NOAA[6] for the Spring Term 2015

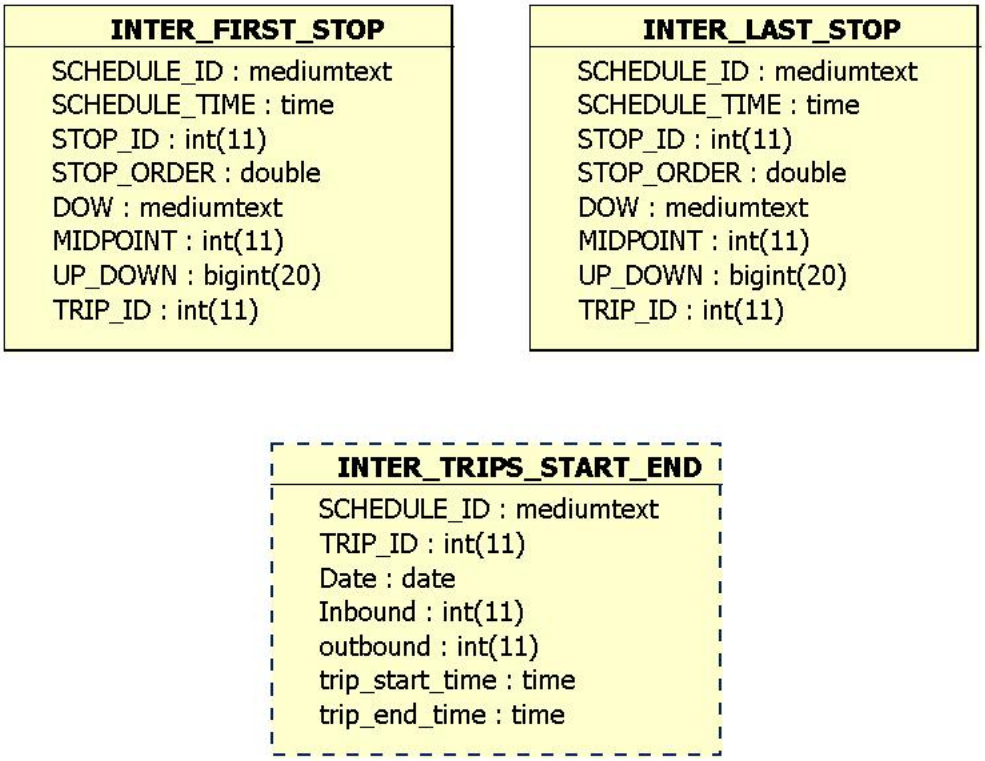
3.2 Base Data Model

The base tables contain the schedule data from both IU and DBMAP but in the format required to calculate the variation of actual trip time from schedule time.

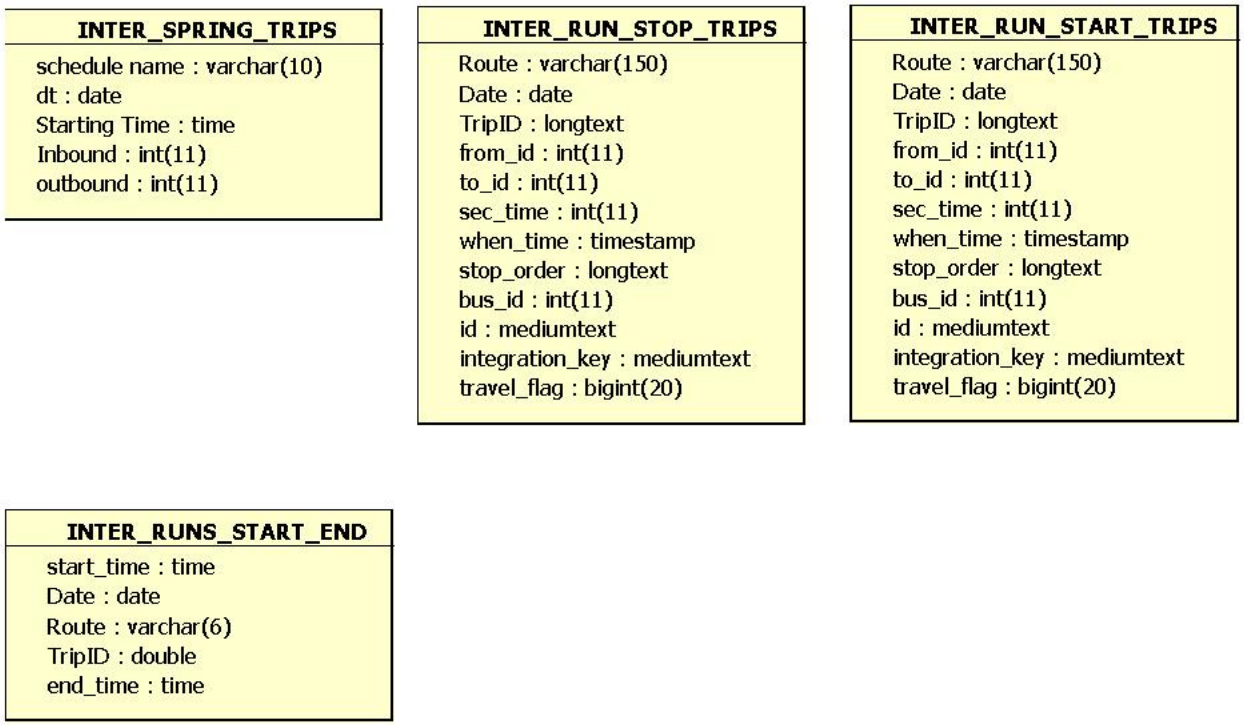


3.3 Intermediate Model

Since one of the primary objective was to find how late/early each trip has run the tables below are created to be populated with trip/schedule times (The method is explained in next section 'Data Preparation ')

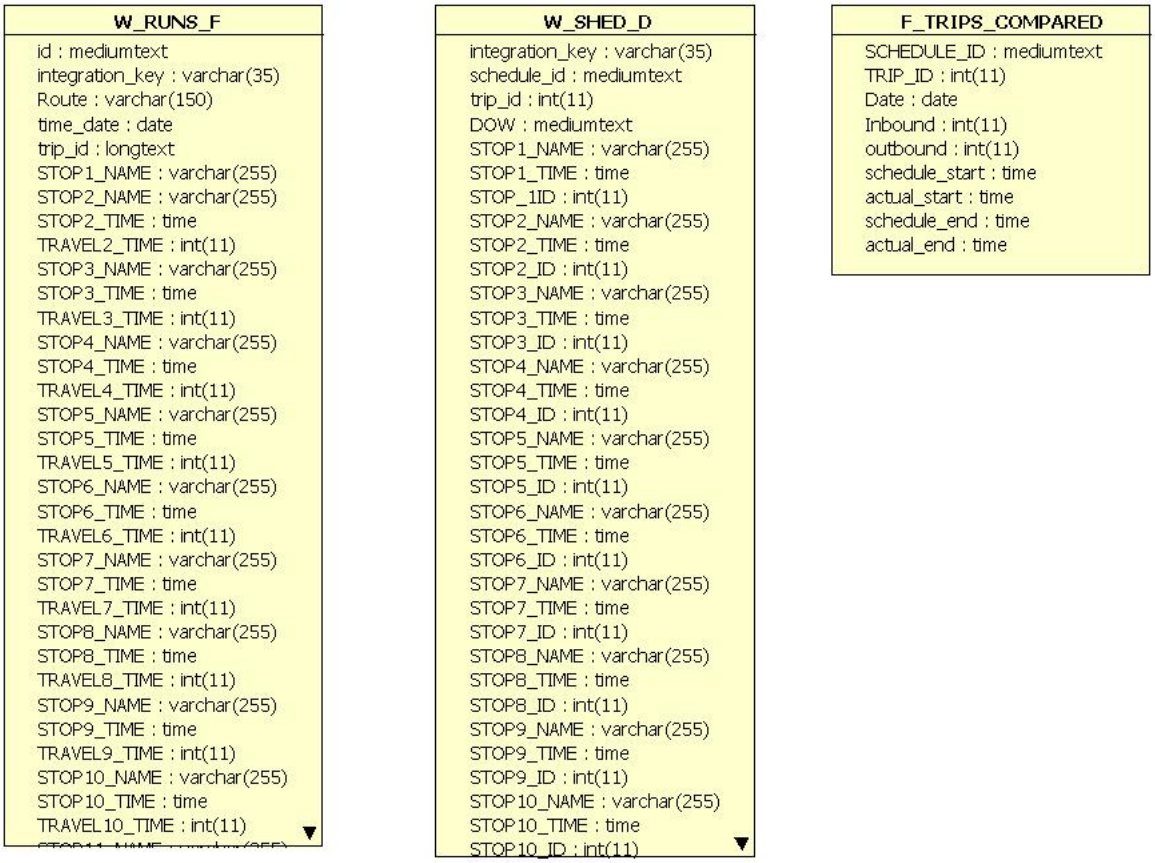


The above tables are intermediate tables that are used in calculating **scheduled** start and end time of each trip



3.4 Data Warehouse Model

The data model by far developed is an OLTP mode l, which is used for operational efficiency. Inorder to facilitate a better analytic needs, these abstract entities are to be converted to a dimension and fact model. However, instead of traditional OLAP model, we have considered that the dimension to be the schedule and run to be facts. They reside in the model as shown below.



It can be seen that instead of a one-row per stop approach, this data model now has all trip data in one row. The data model is designed in such a way that it can hold upto 30 stops and a select statement can be used to query the data from these tables and calculate variance.

3.5 Data Preparation

This section details the process that was undertaken to convert the data obtained from IU and DBMAP into actionable format. The format of data that is considered actionable is the flattened table format noted in the Data Warehouse model section. The data preparation involved the following steps:

1. Generating a TripID for each trip. This is generated by looking at

the stop ID and whenever the originating stop occurs, the trip id is incremented.

2. Consolidating different fields such as RouteID, Busid, stopid, time from multiple tables

The above two steps are performed for both schedule data and actual trip data. Now TripId can be further used to compare scheduled and actual time. As discussed above, the data is now available in OLTP tables with the trip id and stop order. Now asper the data model the data has to be flattened to the new structure.

Route	Stop	Time
Route A	Stop 1	8:00
Route A	Stop 2	8:05
Route A	Stop 3	8:10
Route B	Stop 1	9:00
Route B	Stop 2	9:15

Route	Stop1	Stop 1 Time	Stop 2	Stop 2 Time
Route A	Stop 1	8:00	Stop 2	8:05
Route B	Stop 1	9:00	Stop 2	9:15

Multiple rows per Trip to single row per trip - Flattening Hierarchy

As shown above the transformation is made by repeated joins to the same table. For the schedule table,

```
concat(DOW,concat('~',concat(trip_id,concat('~',Schedule_ID)))) 'integration_key'
```

is used as an integration key. This is nothing but the concatenation of day of week, trip id, and the route to create an unique key for every trip. Thus the data is available on the schedule dimension as below,

integration_key	schedule_id	trip_id	DOW	STOP1_NAME	STOP1_TIME	STOP_1ID	STOP2_NAME	STOP2_TIME	STOP2_ID	STOP3_NAME	STOP3_TIME	STOP3_ID	STOP
M~1~M-A1.1	M-A1.1	1	M	Stadium (A)	07:25:00	67	Well's Library	07:32:00	1	3rd & Jordan	07:34:00	6	IMU
M~2~M-A1.1	M-A1.1	2	M	Stadium (A)	07:49:00	67	Well's Library	07:56:00	1	3rd & Jordan	07:58:00	6	IMU
M~3~M-A1.1	M-A1.1	3	M	Stadium (A)	08:20:00	67	Well's Library	08:27:00	1	3rd & Jordan	08:29:00	6	IMU
M~4~M-A1.1	M-A1.1	4	M	Stadium (A)	08:55:00	67	Well's Library	09:02:00	1	3rd & Jordan	09:04:00	6	IMU
M~5~M-A1.1	M-A1.1	5	M	Stadium (A)	09:47:00	67	Well's Library	09:54:00	1	3rd & Jordan	09:56:00	6	IMU
M~6~M-A1.1	M-A1.1	6	M	Stadium (A)	10:29:00	67	Well's Library	10:36:00	1	3rd & Jordan	10:38:00	6	IMU
M~7~M-A1.1	M-A1.1	7	M	Stadium (A)	11:07:00	67	Well's Library	11:14:00	1	3rd & Jordan	11:16:00	6	IMU

Once the data of schedule is loaded, now the data of the actual runs can be loaded. However to this new table, the unique key is the concatenation of data and the other components as described above.

id	integration_key	RECON_KEY	Route	time_date	trip_id	STOP1_NAME	STOP2_NAME	STOP2_TIME	TRAVEL2_TIME	STOP1_TIME
2015-01-13~T~2~T-B1.1	T~2~T-B1.1	T~3~T-B1.1	T-B1.1	2015-01-13	2	Fisher Court ()	ZBT	08:30:05	352	08:24:29
2015-01-13~T~3~T-B1.1	T~3~T-B1.1	T~4~T-B1.1	T-B1.1	2015-01-13	3	Fisher Court ()	ZBT	08:57:53	252	08:53:57
2015-01-13~T~4~T-B1.1	T~4~T-B1.1	T~5~T-B1.1	T-B1.1	2015-01-13	4	Fisher Court ()	ZBT	09:35:40	623	09:25:32
2015-01-13~T~5~T-B1.1	T~5~T-B1.1	T~6~T-B1.1	T-B1.1	2015-01-13	5	Fisher Court ()	ZBT	10:05:40	338	10:00:20
2015-01-13~T~6~T-B1.1	T~6~T-B1.1	T~7~T-B1.1	T-B1.1	2015-01-13	6	Fisher Court ()	ZBT	10:45:54	1072	10:28:18
2015-01-13~T~7~T-B1.1	T~7~T-B1.1	T~8~T-B1.1	T-B1.1	2015-01-13	7	Fisher Court ()	ZBT	11:23:49	97	11:22:27
2015-01-13~T~8~T-B1.1	T~8~T-B1.1	T~9~T-B1.1	T-B1.1	2015-01-13	8	Fisher Court ()	ZBT	12:05:04	1164	11:45:54

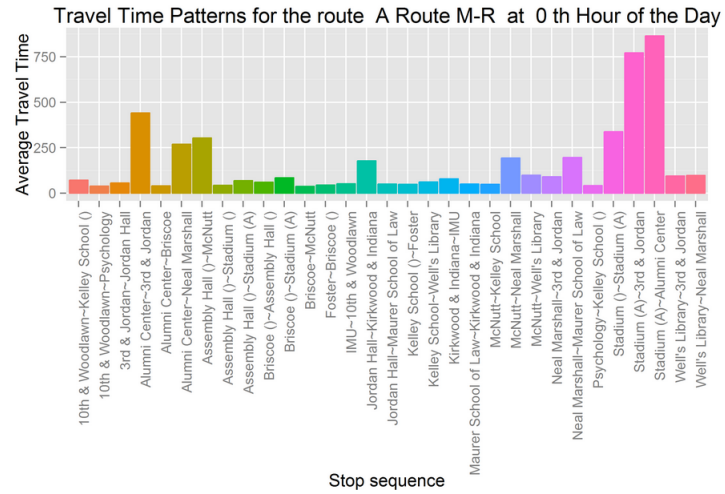
However, even with this, we would not be able to match the schedule and run. It might happen that for a given trip, the bus may not have stopped in stop 1, hence the whole trip would be lost. Hence we calculate an reconciliation key, $RECON_KEY$. It can be seen that these two keys need not be same and the new key can be used to join with the schedule table to calculate variance.

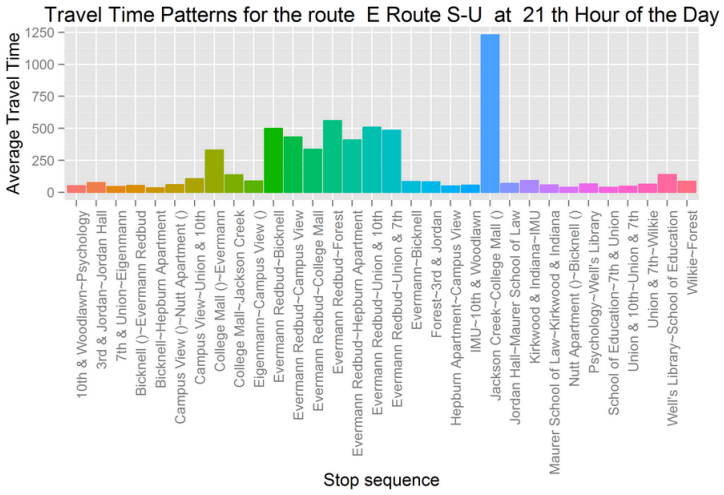
id	integration_key	RECON_KEY	Route	time_date
2015-01-13~T~2~T-B1.1	T~2~T-B1.1	T~3~T-B1.1	T-B1.1	2015-01-13
2015-01-13~T~3~T-B1.1	T~3~T-B1.1	T~4~T-B1.1	T-B1.1	2015-01-13
2015-01-13~T~4~T-B1.1	T~4~T-B1.1	T~5~T-B1.1	T-B1.1	2015-01-13
2015-01-13~T~5~T-B1.1	T~5~T-B1.1	T~6~T-B1.1	T-B1.1	2015-01-13
2015-01-13~T~6~T-B1.1	T~6~T-B1.1	T~7~T-B1.1	T-B1.1	2015-01-13
2015-01-13~T~7~T-B1.1	T~7~T-B1.1	T~8~T-B1.1	T-B1.1	2015-01-13
2015-01-13~T~8~T-B1.1	T~8~T-B1.1	T~9~T-B1.1	T-B1.1	2015-01-13

Now the variance can be calculated directly with just a select query.

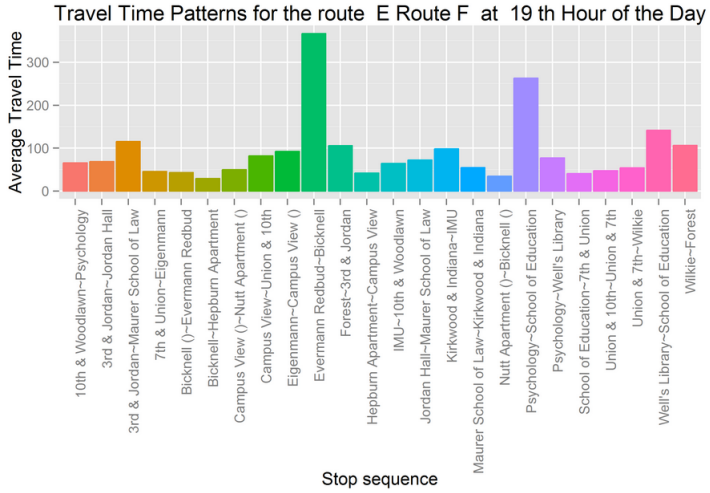
4 Exploratory Analysis

The start of any Data Analytics involves understanding of the data. An initial Exploratory data analysis was made using R and ggplot2 library. For example the below graph indicates the average time between pairs of stops at 12 AM for Route A. It can be seen that the travelling time is very less during this time of the day.

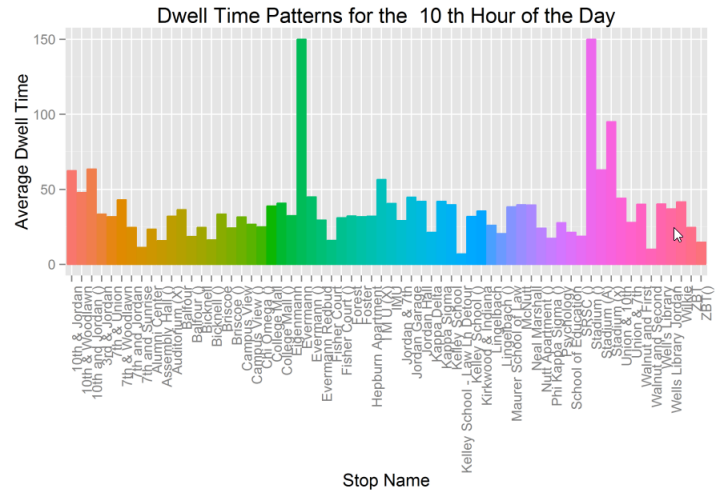




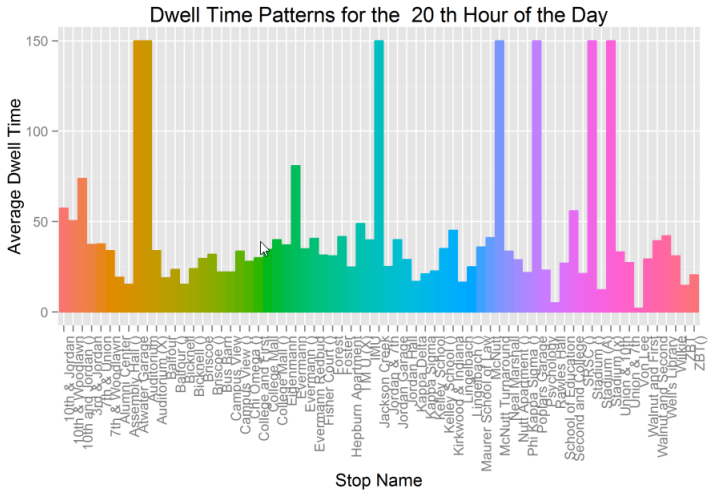
However, a change in trend happens during the ”peak times”. During evening 7, the average traveling time increases drastically, for Eg, 10th and Wood Lawn to psychology takes less than a minute during morning hours, however during evenings, it is doubled. The same trend can be observed in below plots.



However things get different when we look into the dwell time. The average dwell time was plotted for every single hour of the day, it is interesting to note that the maximum time the bus spends is in the originating stops!



Another interesting trend is that as the day progresses, dwell time increases in major stops. There is in fact spike in the major stops like IMU, Kirkwood and Indiana and Kelley's school of Business.

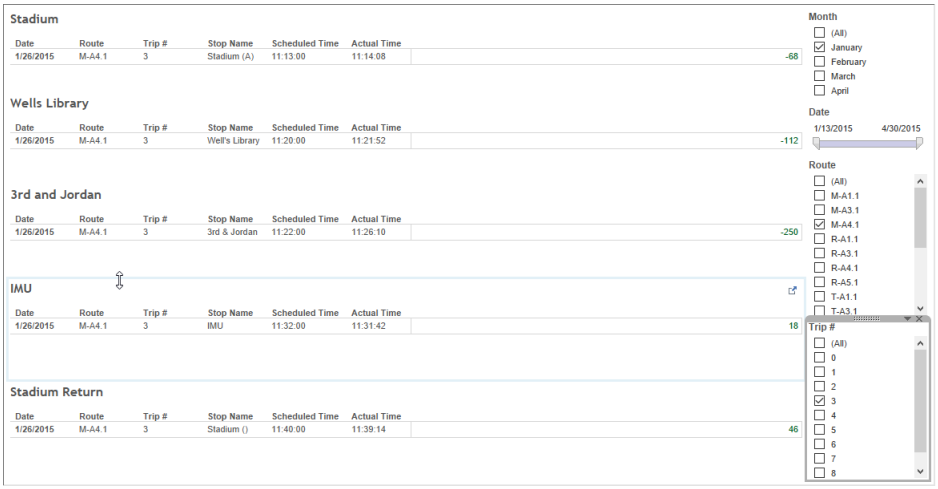


All these graphs and more visualizations are available in the Github Repository, `https://github.com/sarvothaman/nefarious-octo-rutabaga/tree/master/EDA`

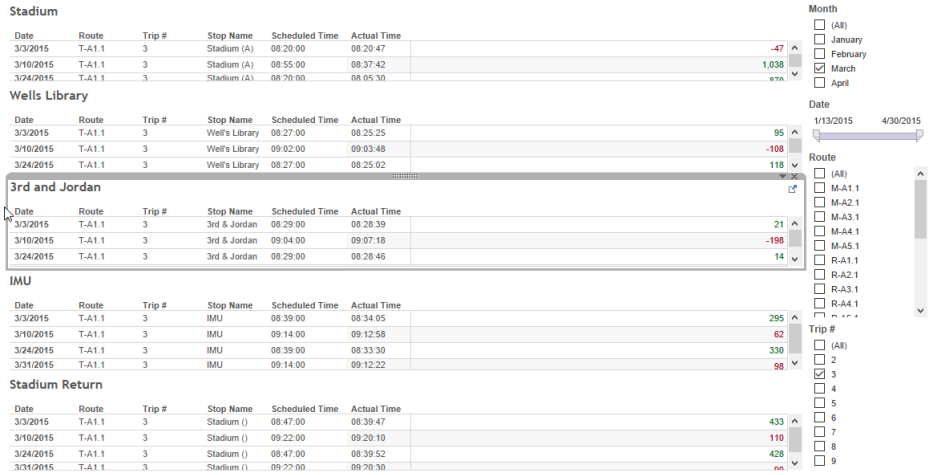
5 Reporting

The Main objective of this case study is to report how consistent the busses run in accordance to the schedule. ie, the time deviation $t_{schedule} - t_{acutal}$ is an important metric. Tableau reporting solution was used to report these findings.

Tableau provides an excellent platform for developing these reports. The variance reports are reports which gives the flexibility to select the month, data, route and trip and check the variance in each stops. This is available on cloud and can be accessed by anyone for free. Below is the screenshot of the application hosted.



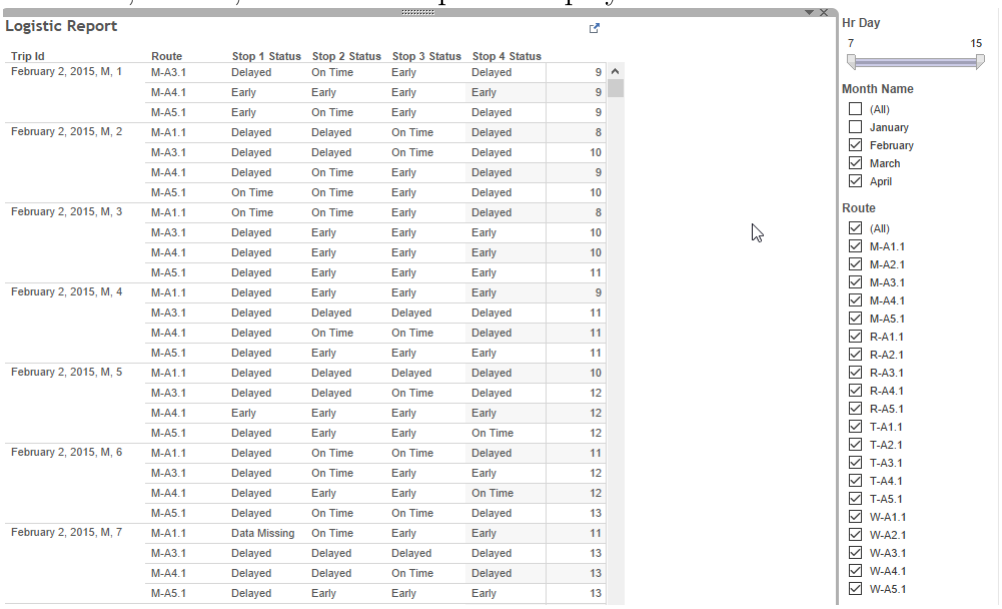
For example, the above shown trip is the 3rd trip of A1 on January 26th 2015. It can be seen that the departure from stadium A is at 11:13, however the bus departed at 11:14 bringing with a minute delay. It is interesting that at 3rd and Jordan, the bus was about 3 minutes late, however it reached IMU one minute early. This suggests that there is still space for optimizing the schedule. Multiple trips can also be selected as shown below to access trip information.



Status Reports

Sometimes we are interested in only seeing whether the bus has arrived on-

time or early to the main stops. The below report is called the logistic report and is available on cloud. This gives the user the option to drill down a specific date, month, route then trip and displays the status.



Logistic Report

Trip Id	Route	Stop 1 Status	Stop 2 Status	Stop 3 Status	Stop 4 Status	
February 2, 2015, M, 1	M-A3.1	Delayed	On Time	Early	Delayed	9
	M-A4.1	Early	Early	Early	Early	9
	M-A5.1	Early	On Time	Early	Delayed	9
February 2, 2015, M, 2	M-A1.1	Delayed	Delayed	On Time	Delayed	8
	M-A3.1	Delayed	Delayed	On Time	Delayed	10
	M-A4.1	Delayed	On Time	Early	Delayed	9
	M-A5.1	On Time	On Time	Early	Delayed	10
February 2, 2015, M, 3	M-A1.1	On Time	On Time	Early	Delayed	8
	M-A3.1	Delayed	Early	Early	Early	10
	M-A4.1	Delayed	Early	Early	Early	10
	M-A5.1	Delayed	Early	Early	Early	11
February 2, 2015, M, 4	M-A1.1	Delayed	Early	Early	Early	9
	M-A3.1	Delayed	Delayed	Delayed	Delayed	11
	M-A4.1	Delayed	On Time	On Time	Delayed	11
	M-A5.1	Delayed	Early	Early	Early	11
February 2, 2015, M, 5	M-A1.1	Delayed	Delayed	Delayed	Delayed	10
	M-A3.1	Delayed	Delayed	On Time	Delayed	12
	M-A4.1	Early	Early	Early	Early	12
	M-A5.1	Delayed	Early	Early	On Time	12
February 2, 2015, M, 6	M-A1.1	Delayed	On Time	On Time	Delayed	11
	M-A3.1	Delayed	On Time	Early	Early	12
	M-A4.1	Delayed	Early	Early	On Time	12
	M-A5.1	Delayed	On Time	On Time	Delayed	13
February 2, 2015, M, 7	M-A1.1	Data Missing	On Time	Early	Early	11
	M-A3.1	Delayed	Delayed	Delayed	Delayed	13
	M-A4.1	Delayed	Delayed	On Time	Delayed	13
	M-A5.1	Delayed	Early	Early	Early	13

For example the Feburary 2nd, A4 route, 1st trip reached all its stop ahead time.

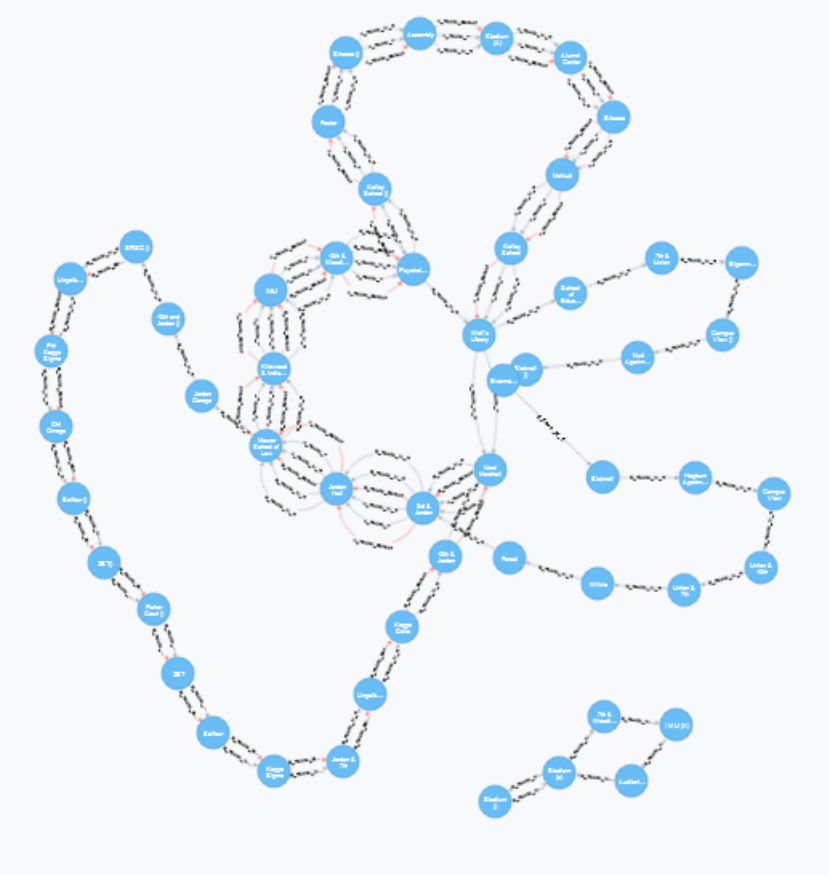
Thus care has been taken during this case study that due importance is given to usability of the output artifacts and can be used by the business people rather than just the esoteric data miners.

6 Visualization

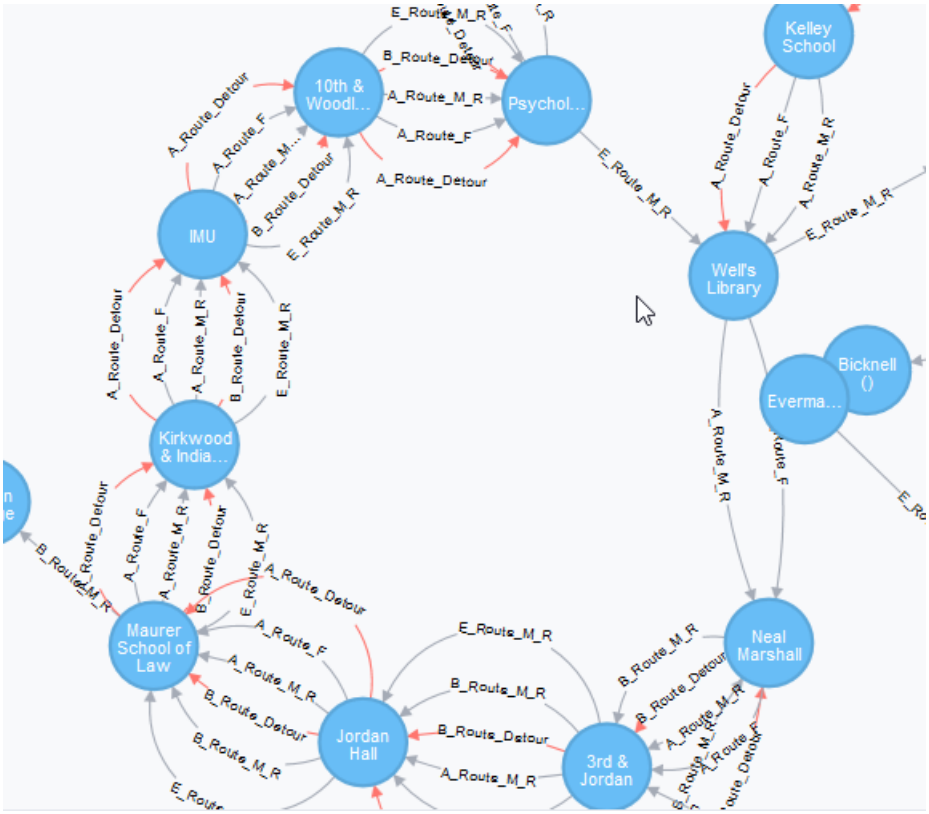
Another aspect of the data is the ability to visualize the data to understand the route. Neo4j is used as the background data store for saving node data. pyMysql module was used to connect to mysql and py2neo was used to connect to the cloud instance.

A python program was written to aggregate the data from MySql and load it to cloud Neo4j instance. Once the entire data is loaded, the graph

looks as follows.



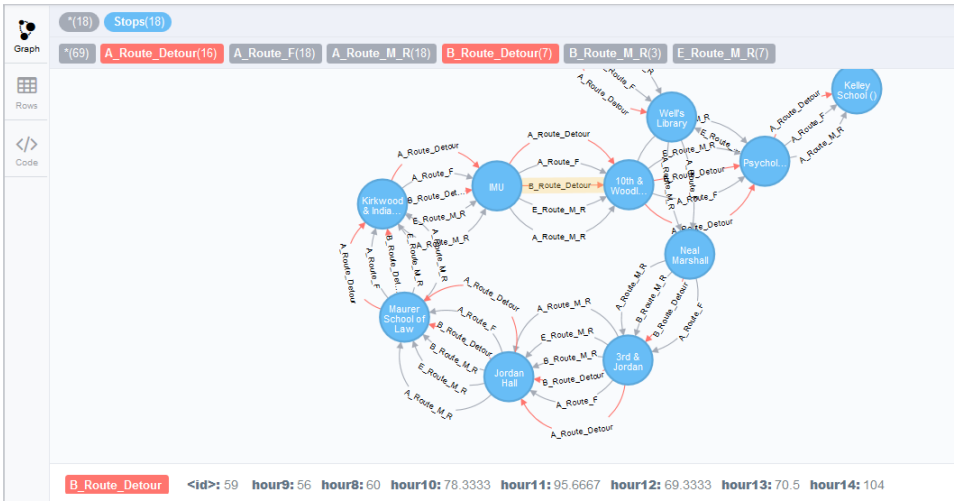
A close in screenshot is as follows. The data model is created in such a way that each stops are represented as nodes and each directed arrow is the route. These edges has a properties like travel time at any given hour of the day and the nodes contain average dwell time in that stop for a given day



Neo4j uses the language called Cypher instead of the SQL query. An example cypher query is as follows

```
MATCH p =(a:Stops { name: "Neal Marshall" })
-[:A_Route_M_R *]-
(b:Stops { name: "IMU" })
RETURN p, length(p)
```

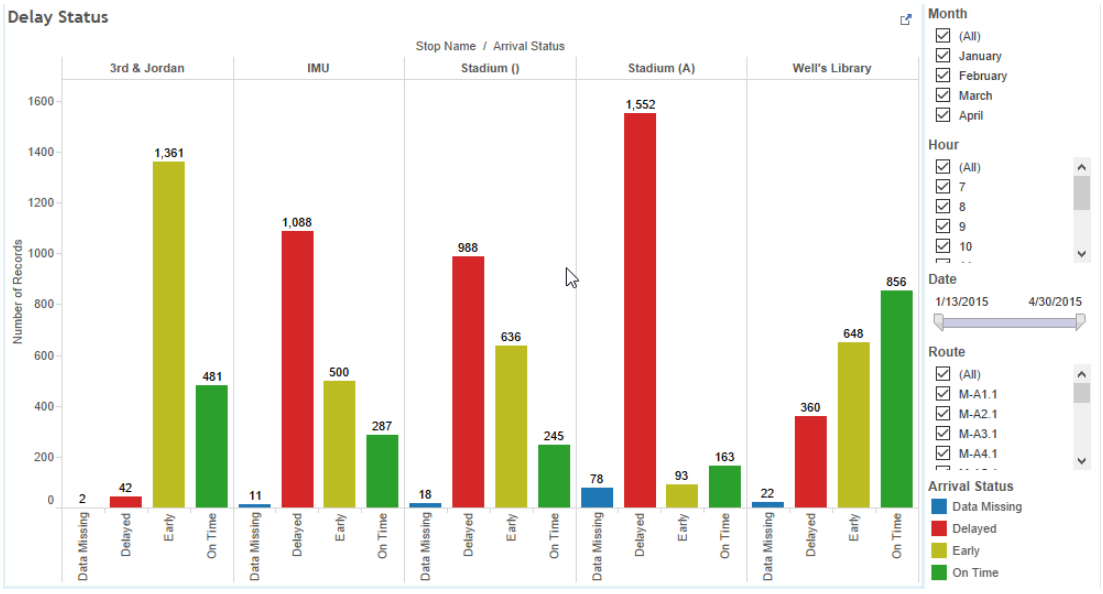
The above cypher selects a path from Neal Marshall to IMU and follows the route A in the Monday to Thursday schedule.



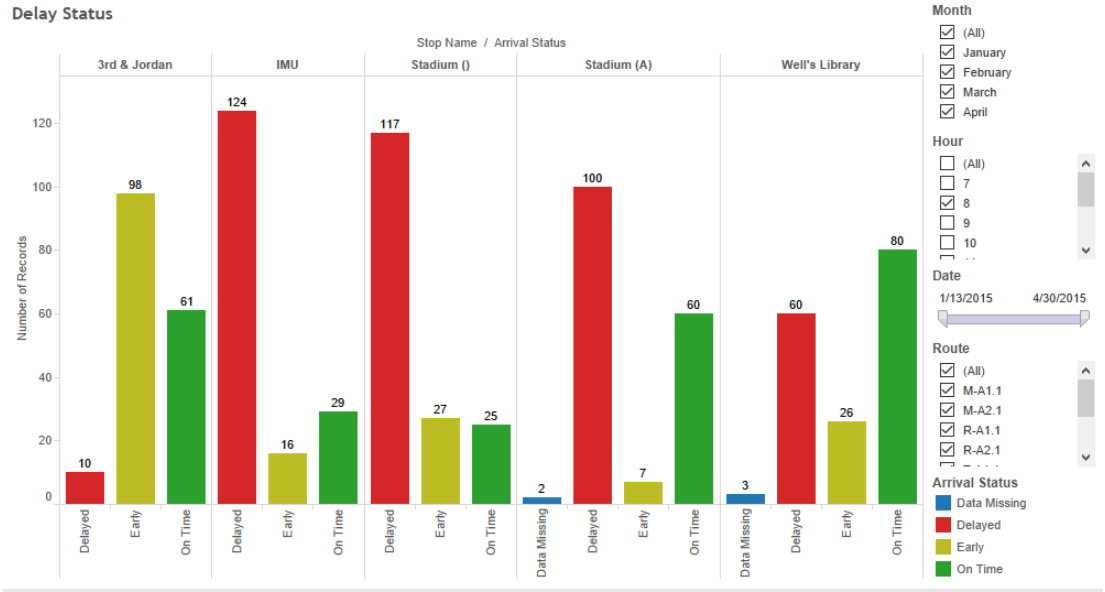
It can be seen below that there are parameters like hour9 indicates the travelling time at 9th hour of the day. This was used to predict the delay/arrival time and most of the time it predicted the time accurate upto Plus or minus three minutes.

Status Reporting

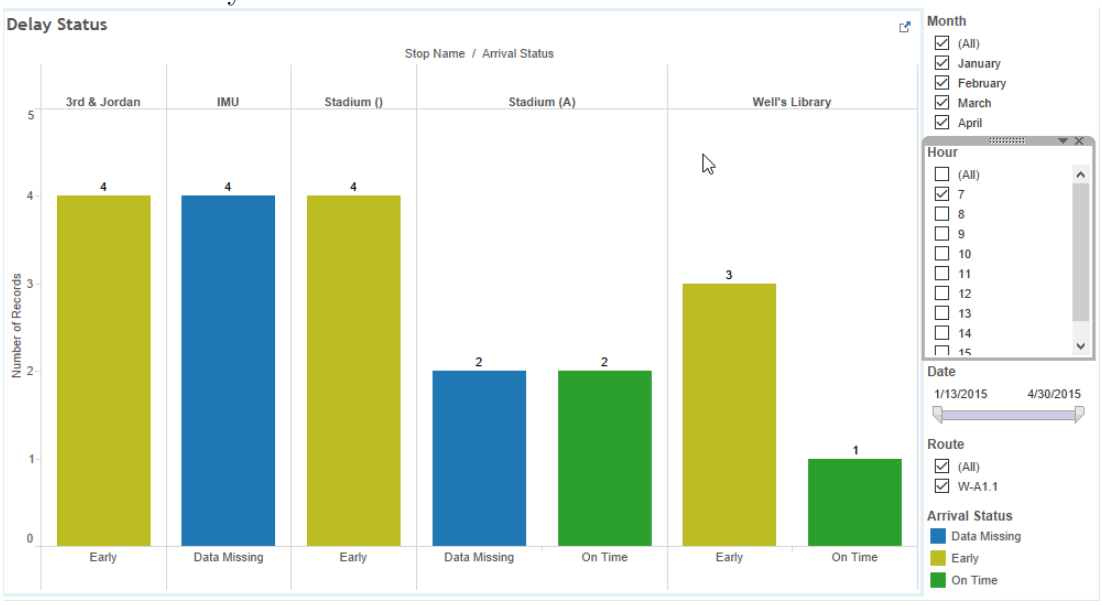
As seen in above reports, for effective decision making the counts of instances bus statuses over a period of time is important. The below plots help in such interpretation,



Eg, the above graph gives the count of the instances of the statuses. Generally it can be seen that stadium is always delayed and wells library is always on time.



This also lets the user to select the hour of the day, month, day, route and see how the delay status is affected.



In refreshing change, During the mornings, for most of the stops the bus reaches either early or on time.

7 Statistical Model

Logistic Regression Model

The data from the MySql database was made longitudinal again and was exported as an CSV file. This file as sourced to the R.

A logistic Regression model was built over this dataset to predict whether the thr A Route would be On-Time, or Delayed or Early.

As described earlier, if the actual time is a minute or earlier than the schedule time, it is marked Early. If the bus arrives a minute before or after the scheduled time, then it is termed as On-Time. If the bus is delayed by more than a minute, it is termed as late.

In essense,
On time : Running as per the scheduled time with a tolerance of plus or minus one minute
Early: Ahead by more than a minute to the scheduled time
Late: delay of more than a minute to the scheduled time

Thus, the logistic model would have 3 classes, On-Time, Early and Delayed.

Model 1 The model was built in such a way that the prediction in first stop would dependant on the time of the hour and the trip of the day.
Model 2 The second model was built in such a way that this would consider the status of arrival in the first stop, hour and the day of travel.
Iterative Models The consecutive models for stops iteratively built on this process, all constant factors and the previous stop status

```
setwd("C:\\Users\\Sarvo\\nefarious-octo-rutabaga\\R Models\\Data")
library(nnet)
routea<-read.csv("DATA.csv")
# Stop 1 Model
stop1.fit<-multinom(Stop.1.Status~Day+Hr.Day,data=routea)

## # weights:  24 (15 variable)
## initial  value 2614.551165
## iter  10 value 1136.297534
## iter  20 value 1087.220212
## final  value 1087.212606
## converged

# Predict Stop 1 Status
stop1.prob<-data.frame(Day=c("M"),Hr.Day=10)
predict(stop1.fit,newdata = stop1.prob,"probs")

## Data Missing      Delayed        Early      On Time
##   0.03082123   0.77910313   0.07317609   0.11689955

#Stop 2 status
stop2.fit<-multinom(Stop.2.Status~Day+Hr.Day+Stop.1.Status,data=routea)
```

```
## # weights:  36 (24 variable)
## initial  value 2614.551165
## iter  10 value 2117.757741
## iter  20 value 2011.097163
## iter  30 value 1998.078206
## iter  40 value 1997.885031
## final  value 1997.884320
## converged

#Prediction for Stop 2 status
stop2.prob<-data.frame(Day=c("M"),Hr.Day=10,Stop.1.Status=c("Delayed"))
predict(stop2.fit,newdata = stop2.prob,"probs")

## Data Missing      Delayed        Early      On Time
##   0.01664346   0.14224930   0.36711064   0.47399660

#Stop 3 Status
stop3.fit<-multinom(Stop.3.Status~Day+
                                                Hr.Day+
                                                Stop.1.Status+
                                                Stop.2.Status,
                                          data=routea)

## # weights:  48 (33 variable)
## initial  value 2614.551165
## iter  10 value 861.842048
## iter  20 value 708.720592
## iter  30 value 685.807742
## iter  40 value 684.351992
## iter  50 value 684.315511
## final  value 684.313809
## converged

#Prediction for Stop 3
stop3.prob<-data.frame(Day=c("M"),Hr.Day=10,
                                              Stop.1.Status=c("Delayed"),
                                              Stop.2.Status=c("On Time"))
predict(stop3.fit,newdata = stop3.prob,"probs")
```

```
## Data Missing      Delayed        Early      On Time
## 3.381334e-03 7.365777e-11 7.682267e-01 2.283920e-01

# Stop 4 Model
stop4.fit<-multinom(Stop.4.Status~Day+
                                                  Hr.Day+
                                                  Stop.1.Status+
                                                  Stop.2.Status+
                                                  Stop.3.Status,
                                              data=routea)

## # weights:  60 (42 variable)
## initial  value 2614.551165
## iter  10 value 1261.966796
## iter  20 value 1136.788612
## iter  30 value 1099.663056
## iter  40 value 1096.691497
## iter  50 value 1096.544668
## iter  60 value 1096.531568
## final  value 1096.531328
## converged

#Prediction
stop4.prob<-data.frame(Day=c("M"),
                                                 Hr.Day=10,
                                                 Stop.1.Status=c("Delayed"),
                                                 Stop.2.Status=c("On Time"),
                                                 Stop.3.Status=c("On Time"))
predict(stop4.fit,newdata = stop4.prob,"probs")

## Data Missing      Delayed        Early      On Time
## 2.966553e-18 9.588480e-01 8.424239e-12 4.115202e-02
```

A keen observer can see that this prediction is in accordance to the results of Tableau Graphs. Even there, the pattern that was observed is that if the bus starts delayed, it comes on-time to Stop 2, and comes Delayed to Stop 3.

Thus this model can give a cateogorical answers to queries about the existing status of the bus.

8 Concluding remarks

Throughout our exploratory analysis and modeling we were able to make observations that are consistent with reality (verified by logic as well as with actual data) as well as make certain predictions (using neo4j model for example). While the results are preliminary and simple in nature, the framework and methods of analysis can be easily scaled to a greater magnitude. We consider this an important step, since ability to quicky analyze the data is critical in achieving the overall objectives

9 Future Work

As stated earlier these analysis and models were created on only a subset of the given data. We would like to scale this to the entire data at hand and perform more interesting and intutive analysis and visualization of results. We also plan to implement a simulation model which would clearly show how and where the deviation from the planned schedule occurs. In this project we have implemented only a logistic regression model for predictions. We would like to implement a varaity of machine learing algorithms and analyse the performance of each model on this data. By doing so we can find the best algorithm for this predictions and use it as a reliable tool see the effects of future changes which might be done to the bus schedule.

10 References

References

[1] Indiana University. http: //www.iubus.indiana.edu/campus_bus/index.html.

[2] University of Virginia. http://www.virginia.edu/parking/uts/.

[3] Boston University. http://www.bu.edu/thebus/.

[4] Odacir A. Neves Marcos E. Cintra. A fuzzy decision tree for bus network management. Universidade Federal Rural do Semi-Arido, Brazil.

[5] Indiana University. http://www.iubus.indiana.edu/campus_bus/bus_schedule.html.

[6] NOAA. http://www.ncdc.noaa.gov/cdo-web/search.