PUBLIC TRANSPORT OPTIMIZATION

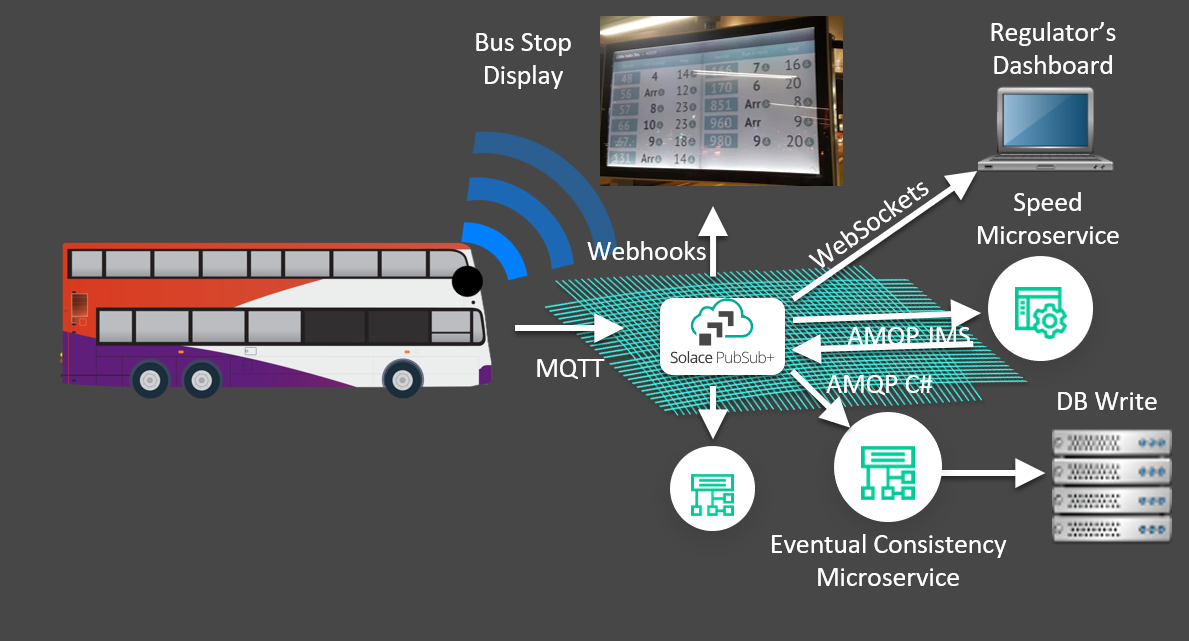
TEAM MEMBER

962121104017: JIINUDAS. M. A

Phase 2 Submission Document

Project: Public Transport Optimization

Introduction:

* Public transport optimization refers to the process of improving the efficiency, accessibility, and sustainability of public transportation systems.
* It involves employing various strategies and technologies to enhance the overall performance of buses, trains, trams, and other modes of public transit.
* This can include route planning, scheduling, fare structures, and the integration of emerging technologies like real-time tracking and data analytics.
* The goal is to make public transport more convenient, cost-effective, and environmentally friendly, ultimately encouraging its use and reducing reliance on private vehicles. 

DriveNo Date and Time Longitude Latitude

Content for Project Phase 2 :

Consider incorporating machine learning algorithms to improve arrival time prediction accuracy based on historical data and traffic conditions.

Data Source

A good data source for public transport optimization using machine learning should be

Accurate, Complete, Covering the geographic area of interest, Accessible

.

Dataset Link: (

<https://www.kaggle.com/datasets/asjad99/rome-taxi-data-subset>)

1 156 2014-02-01 00:00:00.739166+01 41.88367183 12.48777756

2 187 2014-02-01 00:00:01.148457+01 41.92854333 12.46903667

3 297 2014-02-01 00:00:01.220066+01 41.89106861 12.49270456

4 89 2014-02-01 00:00:01.470854+01 41.79317669 12.43212196

5 79 2014-02-01 00:00:01.631136+01 41.90027472 12.46274618

6 191 2014-02-01 00:00:02.048546+01 41.85230476 12.57740658

7 343 2014-02-01 00:00:02.647839+01 41.89217183 12.46969962

8 341 2014-02-01 00:00:02.709888+01 41.91021256 12.47700043

9 260 2014-02-01 00:00:03.458195+01 41.86582086 12.46552211

10 59 2014-02-01 00:00:03.707117+01 41.89678316 12.4821987

11 122 2014-02-01 00:00:04.232912+01 41.92308749 12.50220354

12 311 2014-02-01 00:00:04.436445+01 41.90681379 12.4902084

13 351 2014-02-01 00:00:04.487352+01 41.91005082 12.49660921

14 58 2014-02-01 00:00:05.182068+01 41.91755922 12.51327352

15 196 2014-02-01 00:00:05.429831+01 41.89222982 12.46977921

16 105 2014-02-01 00:00:06.06672+01 41.89714356 12.47295309

17 331 2014-02-01 00:00:06.362172+01 41.90550407 12.44506426

18 362 2014-02-01 00:00:06.508353+01 41.91019934 12.47700165

19 188 2014-02-01 00:00:06.830676+01 41.92193188 12.49078989

20 172 2014-02-01 00:00:07.028304+01 41.91988508 12.50271848

21 352 2014-02-01 00:00:07.040664+01 41.89783253 12.46939475

22 188 2014-02-01 00:00:07.122411+01 41.92266639 12.48712614

23 361 2014-02-01 00:00:07.311678+01 41.9224726 12.48736664

24 321 2014-02-01 00:00:07.629026+01 41.89724661 12.47285113

25 318 2014-02-01 00:00:07.774661+01 41.88323171 12.46921012

26 188 2014-02-01 00:00:07.820636+01 41.92193188 12.49078989

27 317 2014-02-01 00:00:08.452163+01 41.90041222 12.47283687

28 368 2014-02-01 00:00:08.646102+01 41.89045333 12.47419667

29 295 2014-02-01 00:00:09.135615+01 41.89578956 12.47192042

30 197 2014-02-01 00:00:09.207596+01 41.88486123 12.47064281

31 298 2014-02-01 00:00:09.534952+01 41.89379748 12.47038004

32 232 2014-02-01 00:00:09.889075+01 41.90382928 12.48445171

33 315 2014-02-01 00:00:10.098237+01 41.90006536 12.45728872

34 2 2014-02-01 00:00:10.168741+01 41.9081301 12.5043669

35 135 2014-02-01 00:00:10.198107+01 41.93318995 12.51178471

36 248 2014-02-01 00:00:10.709166+01 41.89590199 12.47688189

37 132 2014-02-01 00:00:10.733119+01 41.85528839 12.47714136

38 104 2014-02-01 00:00:10.855133+01 41.79153962 12.2502862

39 234 2014-02-01 00:00:11.214262+01 41.89778328 12.46933121

40 357 2014-02-01 00:00:11.336365+01 41.83459338 12.47166415

41 281 2014-02-01 00:00:11.8629+01 41.89590769 12.48275946

42 341 2014-02-01 00:00:12.098922+01 41.91090884 12.47728158

43 53 2014-02-01 00:00:12.23614+01 41.89120201 12.50254796

44 257 2014-02-01 00:00:12.341827+01 41.92463612 12.4862287

45 37 2014-02-01 00:00:12.578331+01 41.89784189 12.46842041

46 224 2014-02-01 00:00:12.880149+01 41.96531436 12.45640665

47 178 2014-02-01 00:00:12.926514+01 41.92176792 12.48506951

48 174 2014-02-01 00:00:13.311114+01 41.88981282 12.47450704

49 61 2014-02-01 00:00:13.400034+01 41.90031167 12.47273833

50 291 2014-02-01 00:00:13.406692+01 41.85738267 12.49118118

Data Collection and Preprocessing:

* Optimizing public transport involves gathering data on various aspects like passenger demand, traffic patterns, and operational efficiency.
* This data is collected through methods like GPS tracking, passenger surveys, and traffic monitoring.
* Once collected, it's processed to generate insights and inform decision-making.
* Techniques such as data analytics, machine learning, and simulation models can be used to analyze and optimize routes, schedules, and resource allocation for better efficiency and service quality.

Exploratory Data Analysis (EDA):

* Gather data on passenger demand, routes, schedules, vehicle locations, and any other relevant variables.
* Handle missing values: Identify and appropriately deal with missing data points.
* Outlier detection: Address any anomalies that could skew the analysis.
* Calculate mean, median, mode, variance, etc., for key variables.

Feature Engineering:

* Stops and Stations: Distance between stops, location-based demand, connectivity to other transportation modes.
* Time of Day: Different demand patterns during peak hours, off-peak hours, and weekend
* Transfer Efficiency: Ease and efficiency of transferring between different modes of transport.
* Passenger Load Data: Number of passengers on board at any given time.

Advanced Regression Techniques:

* Time Series Regression :Utilize historical data to model the relationship between time-dependent variables (e.g., passenger demand) and other factors such as time of day, day of week, and seasonality.
* Gradient Boosting Regression:Techniques like XGBoost, LightGBM, and CatBoost can handle complex interactions between features and provide accurate predictions.

Model Evaluation and Selection:

* Regression Models: If you're predicting continuous variables like travel time or cost, regression models like linear regression or decision trees could be considered.
* Classification Models: If the problem involves categorical outcomes, such as route selection or mode choice, classification models like Random Forest or Logistic Regression might be appropriate.
* Time Series Models: For forecasting tasks (e.g., predicting demand over time), models like ARIMA or LSTM can be effective.

Model Interpretability:

* Feature Importance Analysis:

Identify which features have the most influence on the model's predictions. Techniques like permutation importance or SHAP (SHapley Additive exPlanations) values can be used.

* \*LIME (Local Interpretable Model-agnostic Explanations)\*:

- LIME creates locally faithful explanations for a specific prediction by training an interpretable surrogate model in the vicinity of that prediction.

* User Feedback and Expert Input:

- Incorporate feedback from stakeholders, domain experts, and end-users to validate and refine the model's interpretations.

. \*Data Preprocessing and Cleaning\*:

- Prepare the data for modeling by handling missing values, outliers, and ensuring data quality.

Deployment and Prediction:

\*Problem Definition\*:

- Clearly define the optimization problem. Are you focusing on route planning, scheduling, fare optimization, or a combination?

\*Data Collection\*:

- Gather relevant data like passenger demand, traffic patterns, schedules, vehicle capacities, and any other factors that influence public transport operations.

Loading the Data

*#load data install.packages("data.table")*

library(data.table)

taxi\_data = fread(file.choose(), header = T, sep = ',',data.table=FALSE)

PROGRAM

Exploring the data:

After loading the dataset, we perform a general exploratory analysis of this dataset to understand the data at hand.

head(dataset, 10)

The dataset contains 4 attributes:

ID of a taxi driver. This is a unique numeric ID.

Date and time in the format Y:m:d H:m:s.msec+tz, where msec is micro-seconds, and tz is a time-zone adjustment.

Longitude, Latitude: These provide explicit trajectory information and if analysed properly contains rich spatiotemporal information.

Since they are a a series of chronologically ordered points, they represent taxi movement traces. i.e spatial trajectory generated by moving taxis in Rome.

In exploratory phase, a good way to start is to get a high level overview of the data using the summary method, which shows the relevant stats such as the total number of samples, possible missing values and the data type for each column. Let's get compute some quick stats(minimum, maximum, and mean location values)

summary(taxi\_data)

Visualisation:

Visualisation is an effective way to understand the data at hand and get a sense of common routes taken by the taxi drivers. Since we are plotting 27 million rows of data, the basic plot will get us nothing but a splash of circles on the screen. We experimeted with ggplot and its size parameter to obtain a sensible. (this also saves compute time).

#Plot the location points (2D plot)

#create a dataframe

taxi\_pickup\_data <- data.frame(taxi\_data[,.(Longitude)], taxi\_data[,.(Latitude)])

taxi\_pickup\_data <- data.frame(taxi\_data[,.(DriveNo)], taxi\_data[,.(DateandTime)])

#basic plot gives a big blob (uncomment if needed)

#plot(taxi\_pickup\_data)

install.packages("ggplot2")

library(ggplot2)

ggplot(taxi\_data, aes(x= Longitude, y= Latitude)) + geom\_point(size=0.02)

This result doesn't make a lot of sense but we can clearly see some points are not part of the trips(outliers). Let's proceed to pre-processing and come back to improving the plot later.

Pre-Processing:

In this step we identify missing values, erroneous entries and remove a few outliers(noise or data that is not relevant to our analysis)

### remove duplicates:

taxi\_data\_subset = taxi\_data\_subset[!duplicated(taxi\_data\_subset[1:3]), ]

### Detecting outliers:

Easiest way to remove outliers is to simply plot the longitude and latitude and visually define the area of Rome on which we want to focus our analysis.

A bounding box was created(as shown in Figure 1) with the below values. The bounding box is based on rome city coordinates from google maps.

Any point outside this bounding box was considered an outlier as it was too far from the heart of the city.

Min\_lat= 41.793710

Max\_lat = 41.991390

Min\_lon = 12.372598

Max\_lon = 12.622537

we will create a Tranformation Function for preprocessing and applying Tranformation to raw data.

nw <- list(lat = 40.5, lon = 13)

se <- list(lat = 43, lon = 11)

trans <- function(x) {

# set coordinates outside of NYC bounding box to NA

ind <- which(x$Longitude < nw$lon & x$Longitude > se$lon)

x$dropoff\_longitude[ind] <- NA

ind <- which(x$Longitude < nw$lon & x$Longitude > se$lon)

x$Longitude[ind] <- NA

}

#Apply the Tranformation

taxi\_data\_subset = trans(taxi\_data\_subset)

The result of applying the bounding box was that 1,563,893 points were removed as outliers.

Now with the relatively clean dataset, we can give visualisation and EDA another shot. For dataset to make more sense, we can visualize the dataset on top of the rome's actual map. We can confirm from the figure below that city centre are the frequently visited regions of rome. We can also see how timings effect the routes as many of the trips early in the day and late afternoon could be part of the daily commute for taxi customers.

Analysis of Travel Behaviour:

We can Obtain the most active, least active, and average activity of the taxi drivers (most time driven, least time driven, and mean time driven)

Identify the most Active drivers:

Most active drivers earn the most revenue and chances are they will be following the most optimal routes instead of engaging in fraudulent behaviour. Let's calculate taxi activity is to compute total time driven by each taxi driver

In [ ]:

compute\_timedriven\_2 <- function(DriverID,mytaxi\_data) {

print("calculating total\_time for driver no.")

print(DriverID)

*#create a dataframe containing only the rows of that taxi driver and selecting DriveNo and Data and Time Columns only*

temp\_dataf <- mytaxi\_data[mytaxi\_data$DriveNo == DriverID,2]

print(summary(temp\_dataf))

if(length(temp\_dataf) == 0){ return(0)

}

*#iterate over the rows total\_time <- as.numeric(1.0000)*

len = length(temp\_dataf) -1 for(i in 1:len){

total\_time = total\_time + as.numeric(difftime(strptime(temp\_dataf[i+1],"%Y-%m-%d %H:%M:%OS"),strptime(temp\_dataf[i],"%Y-%m-%d %H:%M:%OS"),units="min"))

}

print(total\_time) *#return(as.numeric(total\_time), digits=15)*

}

total\_time\_driven <- 0.0000 time\_driven\_list <- list()

for (i in 1:320){

time\_driven\_list[i] <- compute\_timedriven\_2(as.double(i),taxi\_data\_subset) total\_time\_driven <- total\_time\_driven + as.numeric(time\_driven\_list[i])

}

print(which.max(time\_driven\_list)) print(which.min(time\_driven\_list))

*#average time driven print(as.numeric(total\_time\_driven/320),digits = 5)*

we can repurpose this to compute to the time for any given driver.

In [ ]:

compute\_timedriven\_2 <- function(DriverID,mytaxi\_data) {

print("calculating total\_time for driver no.")

print(DriverID)

*#create a dataframe containing only the rows of that taxi driver and selecting DriveNo and Data and Time Columns only*

temp\_dataf <- mytaxi\_data[mytaxi\_data$DriveNo == DriverID,2]

print(summary(temp\_dataf))

if(length(temp\_dataf) == 0){ return(0)

}

*#iterate over the rows total\_time <- as.numeric(1.0000)*

len = length(temp\_dataf) -1 for(i in 1:len){

total\_time = total\_time + as.numeric(difftime(strptime(temp\_dataf[i+1],"%Y-%m-%d %H:%M:%OS"),strptime(temp\_dataf[i],"%Y-%m-%d %H:%M:%OS"),units="min"))

}

print(total\_time) }

compute\_timedriven\_2(211,taxi\_data)

Similarly, we can write another function that lets us compute the total distance travelled by a particular diver:

In [ ]:

linkcode

compute\_distance\_travelled <- function(DriverID,taxi\_data) {

*#create a dataframe containing only the rows of that taxi driver temp\_dataf <- taxi\_data[taxi\_data$DriveNo == DriverID,3:4]*

print(temp\_dataf)

*#radius of earth*

R=6371000

distance = 0

total\_distance = 0 *#print(summary(temp\_dataf))*

len = length(temp\_dataf) -1 for(i in 1:len) {

lon1<- temp\_dataf[i,1] lon2 <- temp\_dataf[i+1,1]

lat1<- temp\_dataf[i,2] lat2 <- temp\_dataf[i+1,2]

dlon = lon2 - lon1 dlat = lat2 - lat1

print(dlon) print(dlat)

a = (sin(dlat/2))^2 + cos(lat1) \* cos(lat2) \* (sin(dlon/2))^2 c = 2 \* atan2( sqrt(a), sqrt(1-a))

distance = R \* c

total\_distance = total\_distance + distance

} return(total\_distance)

} compute\_distance\_travelled(122,taxi\_data\_subset)

OUTPUT:

In the plot above we compare Vfractal values of three different routes taken by three different drivers(extracted in the previous step). The higher values(closer to 2.0) indicate a more torturous path, which means the driver didn’t take the most direct route even though it existed. We repeated the process for several other extracted paths and noticed a similar pattern. The results of our analysis show that taxi drivers may not always take the most optimal route. Moreover, we noticed that Taxi drivers tend to exhibit different behaviour on longer trips, as the incentives change. In particular we noticed that for a longer trip even the most top drivers(with respect to distance travelled) tend to take longer routes.

In this Notebooks we used the rome taxi dataset to analyse pattern movement of taxi drivers. Analysis of this sort can yield various insights for planning bureaus, city planners and transportation analysis etc. After initial exploration and cleaning of data we extracted various routes and then Fractal analysis was employed to quantify tortuosity of movement paths in order to explore how top and ordinary drivers operate on different spatial scales at different times, where the primary focus is to reveal top driver mobility intelligence. Based on our results we can conclude that taxi drivers indeed sometimes try to maximize their income using unethical ways. Moreover, its interesting to see how taxi’ drivers intelligence can be learned from a large number of historical data.

\*\*Future work:\*\* One of the improvements that can made to this work is registering the extracted paths with the underlying road network to get more accurate comparison of paths. We also aim to perform the same analysis based on varying trip length.

CONCLUSION:

Optimizing public transport holds significant potential for enhancing urban mobility, reducing congestion, and mitigating environmental impacts. By employing advanced technologies, implementing efficient routes, and promoting sustainable practices, cities can create a more accessible, reliable, and eco-friendly transportation system. This leads to improved quality of life for residents and fosters economic growth. Continued investment and innovation in public transport optimization are crucial steps towards creating smarter, more livable cities for the future.

FUTURE WORK:

* Embrace advances in automation, artificial intelligence, and data analytics to enhance real-time monitoring, predictive maintenance, and adaptive scheduling.
* Develop seamless connections between different modes of transport, including buses, trains, trams, bicycles, and ride-sharing services, to create a comprehensive, user-friendly transportation network.
* Utilize data-driven insights to tailor services to individual passenger needs, such as customized routes, real-time updates, and fare options.