PUBLIC TRANSPOTATION AND ANALYSIS

PHASE 5: DOCUMENTATION

Abstract:

1. Public transport (also known as public transportation, public transit, mass transit, or simply transit) is a system of transport for passengers by group travel systems available for use by the general public unlike private transport, typically managed on a schedule, operated on established routes, and that charge a posted fee for each trip.
2. Most public transport systems run along fixed routes with set embarkation/disembarkation points to a prearranged timetable, with the most frequent services running to a headway (e.g.: "every 15 minutes" as opposed to being scheduled for any specific time of the day).
3. Urban public transit differs distinctly among Asia, North America, and Europe. In Asia, profit-driven, privately owned and publicly traded mass transit and real estate.

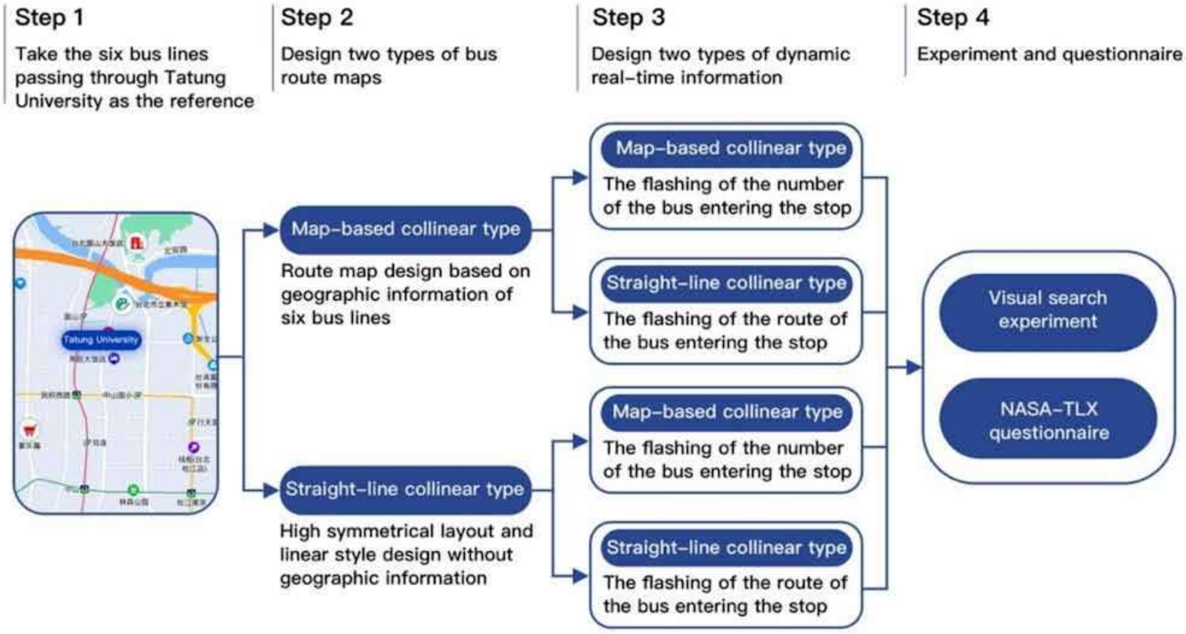
Introduction:

Public transportation systems include a variety of transit options such as buses, light rail, and subways. These systems are available to the general public, may require a fare, and run at scheduled times. The purpose of introducing or expanding public transportation is to increase access to and use of public transit while, at the same time, reducing motor vehicle miles driven and traffic congestion.

Public transportation systems are often implemented at the local or regional level and can be supported by federal initiatives, such as the Fixing America's Surface Transportation (FAST) Act. Los Angeles County is one example of a region that expanded its public transportation system using local, state, and federal funding.

Metropolitan areas have experienced in the last decades an increasing expansion bringing, as a consequence, several socio-economic problems such as an unequal spatial urban development, a high pressure on disposable infrastructure, land and housing shortages, and, with emphasis, lack of urban services. These problems, in addition to low income and unemployment, expel poorer people to urban peripheries where housing costs are lower. But these peripheries are diploid of public services and increase the cost of providing urban infrastructure.

System design:


Step 1
Take the six bus lines passing through Tatung University as the reference
Step 2
Design two types of bus route maps
Step 3
Design two types of dynamic real-time information
Step 4
Experiment and questionnaire
Tatung University
Map-based collinear type
Route map design based on geographic information of six bus lines
Straight-line collinear type
High symmetrical layout and linear style design without geographic information
Map-based collinear type
The flashing of the number of the bus entering the stop
Straight-line collinear type
The flashing of the route of the bus entering the stop
Map-based collinear type
The flashing of the number of the bus entering the stop
Straight-line collinear type
The flashing of the route of the bus entering the stop
Visual search experiment
NASA-TLX questionnaire

Content:

The public bus transportation boarding summary.csv file contains route ,trip,stop and week of year from 20140711.

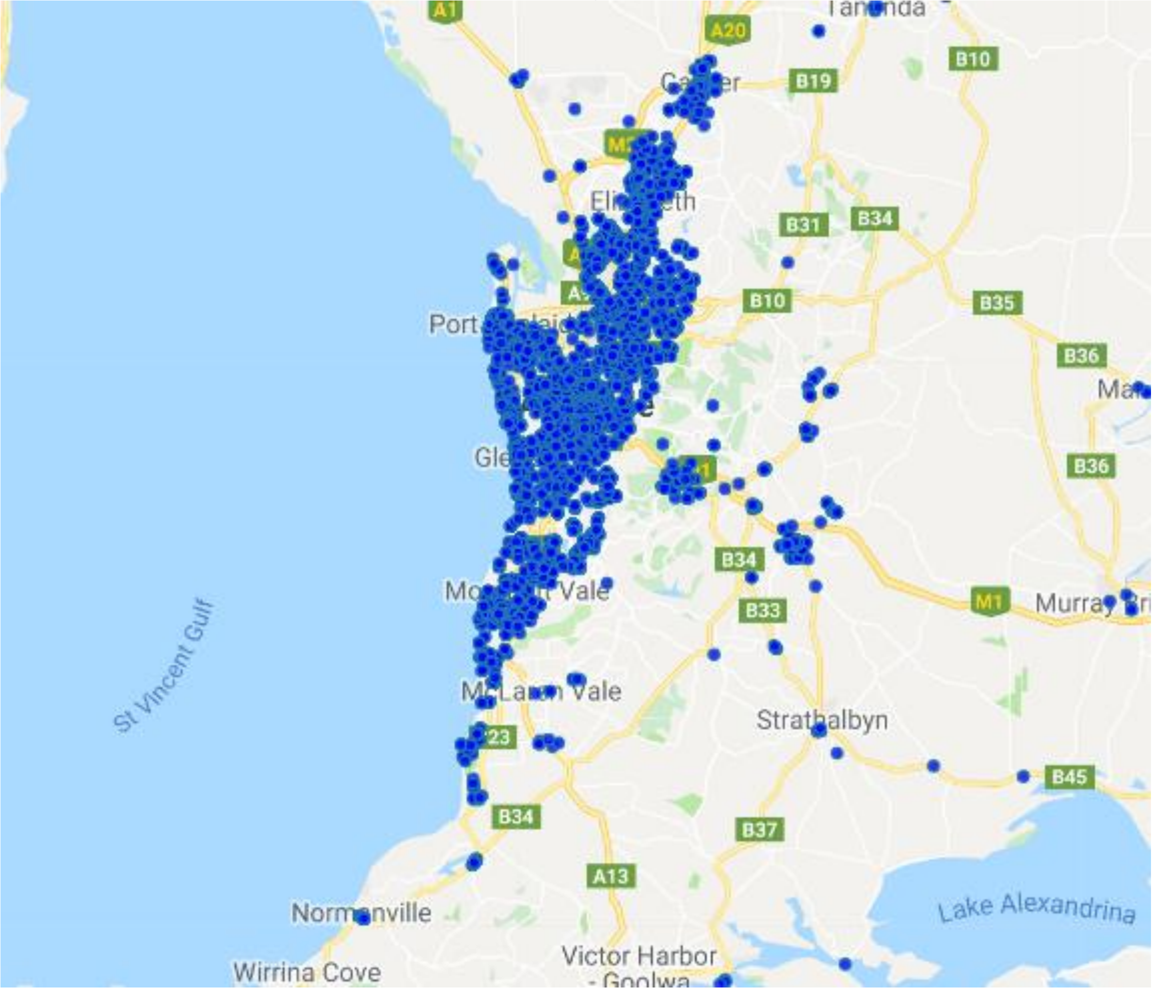
Data source

The data fields in the given file are

- **TripID** Unique identity of trip
- **RouteID** Value representing public transport route
- **StopID** Unique identity of stop
- **StopName** Name of given stop
- **WeekBeginning** Date representing first day of any week
- **NumberOfBoarding** Count of all boarding’s occurred at this stop for the named trip over the previous week

Objective of the notebook:

In this notebook,i have explored how people are travelling from different stops in Adelaide Metropolitan area and the rate at which passengers on each bus route are increasing.finally created a predictive model to find the load of passengers on public Bus transport system in future.



Coding:

In [1]:

```
%matplotlib inline
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import datetime
import os
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler
```

```
import lightgbm as lgb
import xgboost as xgb
from sklearn.metrics import mean_squared_error
from math import sqrt
import warnings
warnings.filterwarnings('ignore')
print(os.listdir("../input/unisys/ptsboardingsummary"))
# Any results you write to the current directory are saved as output.
```

```
['Public Transport Boarding Summary by Route, Trip, Stop and Week of Year.doc',
'20140711.CSV']
```

In [2]:

```
import plotly.plotly as py
import plotly.graph_objs as go
from plotly import tools
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
from bubbly.bubbly import bubbleplot
init_notebook_mode(connected=True)

from bokeh.plotting import figure, save
from bokeh.io import output_file, output_notebook, show
from bokeh.models import ColumnDataSource, GMapOptions,HoverTool
from bokeh.plotting import gmap

import tensorflow as tf
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Input, Dense, GRU,LSTM, Embedding
from tensorflow.python.keras.optimizers import RMSprop
from tensorflow.python.keras.callbacks import EarlyStopping, ModelCheckpoint,
TensorBoard, ReduceLROnPlateau
```

In [3]:

```
## For Multiple Output in single cell
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

In [4]:

```
data = pd.read_csv('../input/unisys/ptsboardingsummary/20140711.CSV')
```

In [5]:

```
out_geo = pd.read_csv('../input/outgeo/output_geo.csv')
route = pd.read_csv('../input/trann11/transit/routes.csv')
```

In [6]:

```
data.shape
data.head(2)
```

Out[6]:

```
(10857234, 6)
```

Out[6]:

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings
0	23631	100	14156	181 Cross Rd	2013-06-30 00:00:00	1
1	23631	100	14144	177 Cross Rd	2013-06-30 00:00:00	1

In [7]:

```
route.head(2)
out_geo.head(2)
```

Out[7]:

	route_id	agency_id	route_short_name	route_long_name	route_desc	route_type	route_url	route_color	route_text_color	RouteGroup
0	100	5	100	Arndale Centre Interchange to Glen Osmond	via Woodville Road, Holbrooks Road,	3	http://www.adelaidemetro.com.au/routes/100	0033CC	ffffff	100-101

	route_id	agency_id	route_short_name	route_long_name	route_desc	route_type	route_url	route_color	route_text_color	RouteGroup
					Marion Roa...					
1	100B	5	100B	Arndale Centre Interchange / Urrbrae Agricultu...	via Kingswood, Hawthorn, Edwardstown, North Pl...	3	http://www.adelaidemetro.com.au/routes/100B	0033CC	ffffff	100-101

Out[7]:

	accuracy	formatted_address	google_place_id	input_string	latitude	longitude	number_of_results	postcode	status	type
0	ROOFTOP	181 Cross Rd, Westbourne Park SA	ChIJKT7I9rbPsGoRVHMHkly-Oyk	181 Cross Rd	-34.966656	138.592148	1	5041	OK	street_address

	accuracy	formatted_address	google_place_id	input_string	latitude	longitude	number_of_results	postcode	status	type
		5041, Australia								
1	ROOFTOP	177 Cross Rd, Westbourne Park SA 5041, Australia	ChIJ-VFZ87bPsGoRyfVgC5qbPpE	177 Cross Rd	-34.966607	138.592301	1	5041	OK	street_address

External Features

Some Important external data fields calculation

- **IsHoliday** Number of public holidays within that week
- **DistanceFromCentre** Distance measure from the city centre

For Calculating Distance between centre with other bus stops by using Longitude and Latitude we have used the Haversine formula

In [8]:

```
from math import sin, cos, sqrt, atan2, radians
def calc_dist(lat1,lon1):
    ## approximate radius of earth in km
    R = 6373.0
    dlon = radians(138.604801) - radians(lon1)
    dlat = radians(-34.921247) - radians(lat1)
    a = sin(dlat / 2)**2 + cos(radians(lat1)) * cos(radians(-34.921247)) * sin(dlon / 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))
    return R * c
```

In [9]:

```
out_geo['dist_from_centre'] = out_geo[['latitude','longitude']].apply(lambda x:
calc_dist(*x), axis=1)
```

In [10]:

```
##Fill the missing values with mode
out_geo['type'].fillna('street_address',inplace=True)
out_geo['type'] = out_geo['type'].apply(lambda x: str(x).split(',')[-1])
```

In [11]:

```
out_geo['type'].unique()
```

Out[11]:

```
array(['street_address', 'transit_station', 'premise', 'political',
       'school', 'route', 'intersection', 'point_of_interest',
       'subpremise', 'real_estate_agency', 'university', 'travel_agency',
       'restaurant', 'supermarket', 'store', 'post_office'], dtype=object)
```

Adding the details regarding the Public holidays from June 2013 to June 2014

In [12]:

```
'''Holidays--
```

Out[12]:

```
"Holidays--\n2013-09-01,Father's Day\n2013-10-07,Labour day\n2013-12-25,Christmas day\n2013-12-26,Proclamation Day\n2014-01-01,New Year\n2014-01-27,Australia Day\n2014-03-10,March Public Holiday\n2014-04-18,Good Friday\n2014-04-19,Easter Saturday\n2014-04-21,Easter Monday\n2014-04-25,Anzac Day\n2014-06-09,Queen's Birthday"
```

In [13]:

```
def holiday_label (row):
    if row == datetime.date(2013, 9, 1) :
          return '1'
    if row == datetime.date(2013, 10, 6) :
          return '1'
    if row == datetime.date(2013, 12, 22) :
          return '2'
    if row == datetime.date(2013, 12, 29):
          return '1'
    if row  == datetime.date(2014, 1, 26):
          return '1'
    if row == datetime.date(2014, 3, 9):
          return '1'
    if row == datetime.date(2014, 4, 13) :
          return '2'
    if row == datetime.date(2014, 4, 20):
        return '2'
    if row == datetime.date(2014, 6, 8):
        return '1'
    return '0'
```

In [14]:

```
data['WeekBeginning'] = pd.to_datetime(data['WeekBeginning']).dt.date
```

In [15]:

```
data['holiday_label'] = data['WeekBeginning'].apply (lambda row: holiday_label(row))
```

Data Aggregation

Combine the Geolocation,Routes and main input file to get final Output File.

In [16]:

```
data= pd.merge(data,out_geo,how='left',left_on = 'StopName',right_on =
'input_string')
```

In [17]:

```
data = pd.merge(data, route, how='left', left_on = 'RouteID', right_on = 'route_id')
```

Columns to keep for further analysis

In [18]:

```
col = ['TripID', 'RouteID', 'StopID', 'StopName',
'WeekBeginning','NumberOfBoardings','formatted_address',
       'latitude',
'longitude','postcode','type','route_desc','dist_from_centre','holiday_label']
```

In [19]:

```
data = data[col]
```

In [20]:

```
##saving the final dataset
data.to_csv('Weekly_Boarding.csv',index=False)
```

In [21]:

```
## getting the addresses for geolocation api.
# Address data['StopName'].unique()
# sub = pd.DataFrame({'Address': Address})
# sub=sub.reindex(columns=["Address"])
# sub.to_csv('addr.csv')
```

Aggregate the Data According to Weeks and Stop names

- **NumberOfBoardings_sum** Number of Boardings within particular week for each Bus stop
- **NumberOfBoardings_count** Number of times data is recorded within week

- **NumberOfBoardings_max** Maximum number of boarding done at single time within week

In [22]:

```
# st_week_grp1 = pd.DataFrame(data.groupby(['StopName','WeekBeginning','type']).agg({'NumberOfBoardings': ['sum', 'count']})).reset_index()
grouped = data.groupby(['StopName','WeekBeginning','type']).agg({'NumberOfBoardings': ['sum', 'count','max']})
grouped.columns = ["_".join(x) for x in grouped.columns.ravel()]
```

In [23]:

```
st_week_grp = pd.DataFrame(grouped).reset_index()
st_week_grp.shape
st_week_grp.head()
```

Out[23]:

```
(207864, 6)
```

Out[23]:

	StopName	WeekBeginning	type	NumberOfBoardings_sum	NumberOfBoardings_count	NumberOfBoardings_max
0	1 Anzac Hwy	2013-06-30	street_address	1003	378	51
1	1 Anzac Hwy	2013-07-07	street_address	783	360	28
2	1 Anzac Hwy	2013-07-14	street_address	843	343	45
3	1 Anzac Hwy	2013-07-21	street_address	710	356	28

	Stop Name	WeekBeginning	type	NumberOfBoardings_sum	NumberOfBoardings_count	NumberOfBoardings_max
4	1 Anzac Hwy	2013-07-28	street_address	898	379	41

Gathering only the Stop Name which having all 54 weeks of Data

In [24]:

```
st_week_grp1 = pd.DataFrame(st_week_grp.groupby('StopName')['WeekBeginning'].count()).reset_index()
```

In [25]:

```
aa=list(st_week_grp1[st_week_grp1['WeekBeginning'] == 54]['StopName'])
```

In [26]:

```
bb = st_week_grp[st_week_grp['StopName'].isin(aa)]
```

In [27]:

```
## save the aggregate data
bb.to_csv('st_week_grp.csv', index=False)
```

Data Exploration

Having Total of 4165 Stops in South Australian Metropolitan Area.

In [28]:

```
data.nunique()
```

Out[28]:

```
TripID                39282
RouteID                 619
StopID                 7397
StopName               4165
WeekBeginning            54
NumberOfBoardings       400
formatted_address      3242
latitude               3029
longitude              3008
postcode                207
type                     16
route_desc              440
```

```
dist_from_centre      3033
holiday_label            3
dtype: int64
```

In [29]:

```
data.shape
data.columns
data.head(3)
```

Out[29]:

```
(10857234, 14)
```

Out[29]:

```
Index(['TripID', 'RouteID', 'StopID', 'StopName', 'WeekBeginning',
       'NumberOfBoardings', 'formatted_address', 'latitude', 'longitude',
       'postcode', 'type', 'route_desc', 'dist_from_centre', 'holiday_label'],
      dtype='object')
```

Out[29]:

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings	formatted_address	latitude	longitude	postcode	type	route_desc	dist_from_centre	holiday_label
0	23631	100	14156	181 Cross Rd	2013-06-30	1	181 Cross Rd, Westbourne Park SA 5041, Australia	-34.966656	138.592148	5041	street_address	via Woodville Road, Holbrooks Road, Marion Roa...	5.180961	0

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings	formatted_address	latitude	longitude	postcode	type	route_desc	dist_from_centre	holiday_label
1	23631	100	14144	177 Cross Rd	2013-06-30	1	177 Cross Rd, Westbourne Park SA 5041, Australia	-34.966607	138.5923301	5041	street_address	via Woodville Road, Holbrooks Road, Marion Roa...	5.172525	0
2	23632	100	14132	175 Cross Rd	2013-06-30	1	175 Cross Rd, Westbourne Park SA 5041, Australia	-34.966758	138.592715	5041	street_address	via Woodville Road, Holbrooks Road, Ma	5.180709	0

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings	formatted_address	latitude	longitude	postcode	type	route_desc	dist_from_centre	holiday_label
												rion Roa...		

In [30]:

```
data.isnull().sum()
```

Out[30]:

```
TripID                     0
RouteID                    0
StopID                     0
StopName                   0
WeekBeginning              0
NumberOfBoardings          0
formatted_address       3506
latitude                   0
longitude                  0
postcode              425081
type                       0
route_desc           2106618
dist_from_centre           0
holiday_label              0
dtype: int64
```

Conclusion:

1. The flip side is that the subway rail runs parallel to the commuter rail network that might affect the ridership. Likewise, there is no settlement on the left side after Perungalathur.
2. Due to existing structures and plans for elevated expressway from Tambaram to Chengalpet, the biggest challenge will be to take the corridor to second level. The path which starts from the airport to Perangulathur via Pallavaram, Hasthinapuram, Chitlapakkam, Selaiyur, Tambaram and Tambaram West is a 20 km stretch where the acquisition of land will be maximum.

3. Therefore, this chapter provides better area boundaries and potential destinations designs are provided. Metro projects will be planned and built in isolation in the future, with little regard for feeder trips or other means of transportation.