ARRIVAL TIME PREDICTION USING MACHINE LEARNING

BATCH MEMBER

911921104009 : RAJ THILAK.R

Phase 3 submission document

Project Title : Public Transport Optimization

Phase 3 : Development Part 1

Topic : Start building the arrival time prediction model by loading and preprocessing the dataset.

Public Transport Optimization

Introduction:

❖ Public transportation efficiency is essential for the economicgrowth and sustainability of urban areas. However, indeveloping countries, there are little or no advancements beingmade in this sector. The sizable capacity of public transportvehicles results in fewer trips and reduced fuel consumption.

❖This has led tothe recent trend and drive to modernize public transportationbyintegrating it with the latest technologies for more viableand eco-friendly environments.However, this trend has onlybeen more prevalent in developed countries around the world and not so much in still developing countries.

❖ Commuters - mostly in developing countries - are often leftstranded at pickup spots and clueless about the availability andproximity of their mode of the public vehiclehence the badstigma public transport has This results from theunavailability of real-time information to commuters, poorlymanaged fleets, varying demands, traffic congestion and rigidschedules. Rapid population growth and urbanization lead tooverwhelming travel demands

❖Rapid population growth and urbanization lead tooverwhelming travel demands . A better understandingof the commuters' behaviour helps transport operators estimate the demand and propose better services to improve the commuting experience.

Data Set:

Date	Time	Junction	Vehicles	ID
2015-11-01	00:00:00	1	15	20151101001
2015-11-01	01:00:00	1	13	20151101011
2015-11-01	02:00:00	1	10	20151101021
2015-11-01	03:00:00	1	7	20151101031
2015-11-01	04:00:00	1	9	20151101041
2015-11-01	05:00:00	1	6	20151101051

2015-11-01	06:00:00	1	9	20151101061
2015-11-01	07:00:00	1	8	20151101071
2015-11-01	08:00:00	1	11	20151101081
2015-11-01	09:00:00	1	12	20151101091
2015-11-01	10:00:00	1	15	20151101101
2015-11-01	11:00:00	1	17	20151101111
2015-11-01	12:00:00	1	16	20151101121

Neccesary steps to follow:

1 . Acquire the dataset :

Acquiring the dataset is the first step in data preprocessing in machine learning. To build and develop Machine Learning models, you must first acquire the relevant dataset. This dataset will be comprised of data gathered from multiple and disparate sources which are then combined in a proper format to form a dataset. Dataset formats differ according to use cases. For instance, a business dataset will be entirely different from a medical dataset. While a business dataset will contain relevant industry and business data, a medical dataset will include healthcare-related data.

2. Import all the crucial libraries :

The predefined Python libraries can perform specific data preprocessing jobs. Importing all the crucial libraries is the second step in data preprocessing in machine learning.

Program:

```
%matplotlib inline
        import numpy as np
        import pandas as pd
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
```

3.Import the dataset :

In this step, you need to import the datasets that you have gathered for the ML project at hand. Importing the dataset is one of the important steps in data preprocessing in machine learning.

Program :

```
dataset = pd.read_csv(r"../input/preprocessingdataset/Data.csv")
df = pd.DataFrame(dataset)
```

4. Exploratory Data Analysis (EDA) :

In data preprocessing, it is pivotal to identify and correctly handle the missing values, failing to do this, you might draw inaccurate and faulty conclusions and inferences from the data. Needless to say, this will hamper your ML project.

some typical reasons why data is missing:

A. User forgot to fill in a field.

B. Data was lost while transferring manually from a legacy database.

C. There was a programming error.

D. Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

Basically, there are two ways to handle missing data:

Deleting a particular row – In this method, you remove a specific row that has a null value for a feature or a particular column where more than 75% of the values are missing. However, this method is not 100% efficient, and it is recommended that you use it only when the dataset has adequate samples. You must ensure that after deleting the data, there remains no addition of bias. Calculating the mean – This method is useful for features having numeric data like age, salary, year, etc. Here, you can calculate the mean, median, or mode of a particular feature or column or row that contains a missing value and replace the result for the missing value. This method can add variance to the dataset, and any loss of data can be efficiently negated. Hence, it yields better results compared to the first method (omission of rows/columns). Another way of approximation is through the deviation of neighbouring values. However, this works best for linear data.

4. Feature Engineering :

Depending on your dataset, you may need to create new features or transform existing ones. This can involve one-hot encoding categorical variables, handling date/time data, or scaling numerical features.

6.Splitting the dataset :

Splitting the dataset is the next step in data preprocessing in machine learning. Every dataset for Machine Learning model must be split into two separate sets – training set and test set.

7. Feature scaling :

Feature scaling marks the end of the data preprocessing in Machine Learning. It is a method to standardize the independent variables of a dataset within a specific range. In

other words, feature scaling limits the range of variables so that you can compare them on common grounds.

Another reason why feature scaling is applied is that few algorithms like gradient descent converge much faster with feature scaling than without it.

Importance of loading and processing dataset:

Loading and preprocessing the dataset is an important first step inbuilding any machine learning model. However, it is especiallyimportant for house price prediction models, as house price datasets areoften complex and noisy.

By loading and preprocessing the dataset, we can ensure that the machinelearning algorithm is able to learn from the data effectively and accurately.

Challenges involved in loading and preprocessing a public tansport dataset :

There are a number of challenges involved in loading and preprocessing a public transport dataset, including:

➢ Handling missing values:

Public transport datasets often contain missing values, which can be due to a variety of factors, such as time error or incomplete data collection. Common methods for handling missing values includedropping the rows with missing values, imputing the missing values withthe mean or median of the feature, or using a more sophisticated method such as multiple imputation.

➢ Encoding categorical variables:

Public transport datasets often contain categorical features, such as the type of vehicles, the traffic conditions, and the no of ways. These features need to be encoded before they can be used by machine learning models. One common way to encode categorical

variables is to use one-hot encoding.

➢ Scaling the features:

It is often helpful to scale the features before training a machine learning model. This can help to improve the performance of the model and make it more robust to outliers. There are a variety of ways to scale the features, such as min-max scaling and standard scaling.

➢ Splitting the dataset into training and testing sets:

Once the data has been pre-processed, we need to split the dataset into training and testing sets. The training set will be used to train the model, and the testing set will be used to evaluate the performance of the model on unseen data. It is important to split the dataset in a way that is representative of the real world distribution of the data.

How to overcome the challenges of loading and preprocessing a public transport dataset:

There are a number of things that can be done to overcome the challenges of loading and preprocessing a public transport dataset, including:

➢ Use a data preprocessing library:

There are a number of libraries available that can help with data preprocessing tasks, such as handling missing values, encoding categorical variables, and scaling the features.

➢ Carefully consider the specific needs of your model:

The best way to preprocess the data will depend on the specific machine learning algorithm that you are using. It is important to carefully consider the requirements of the algorithm and to preprocess the data in a way that is compatible with the algorithm.

➢ Validate the preprocessed data:

It is important to validate the preprocessed data to ensure that it is in a format that can be used by the machine learning algorithm and that it is of high quality. This can be done by inspecting the data visually or by using statistical methods.

1.Loading the dataset:

✓ Loading the dataset using machine learning is the process of bringing the data into the machine learning environment so that it can be used to train and evaluate a model.

✓ The specific steps involved in loading the dataset will vary depending on the machine learning library or framework that is being used. However, there are some general steps that are common to most machine learning frameworks:

a.Identify the dataset:

The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.

b.Load the dataset:

Once you have identified the dataset, you need to load it into the machine learning environment. This may involve using a built-in function in the machine learning library, or it may involve writing your own code.

c.Preprocess the dataset:

Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format, and splitting the data into training and test sets.

Program :

In [1] :

```
## import package
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
import tensorflow
from statsmodels.tsa.stattools import adfuller
from sklearn.preprocessing import MinMaxScaler
from tensorflow import keras
from keras import callbacks
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Conv2D, Flatten, Dense, LSTM, Dropout,
GRU, Bidirectional
from tensorflow.keras.optimizers import SGD
import math
from sklearn.metrics import mean_squared_error
import warnings
warnings.filterwarnings("ignore")
```

In [2] :

```
## loading data
df = pd.read_csv("../input/traffic-prediction-dataset/traffic.csv")
df.head()
```

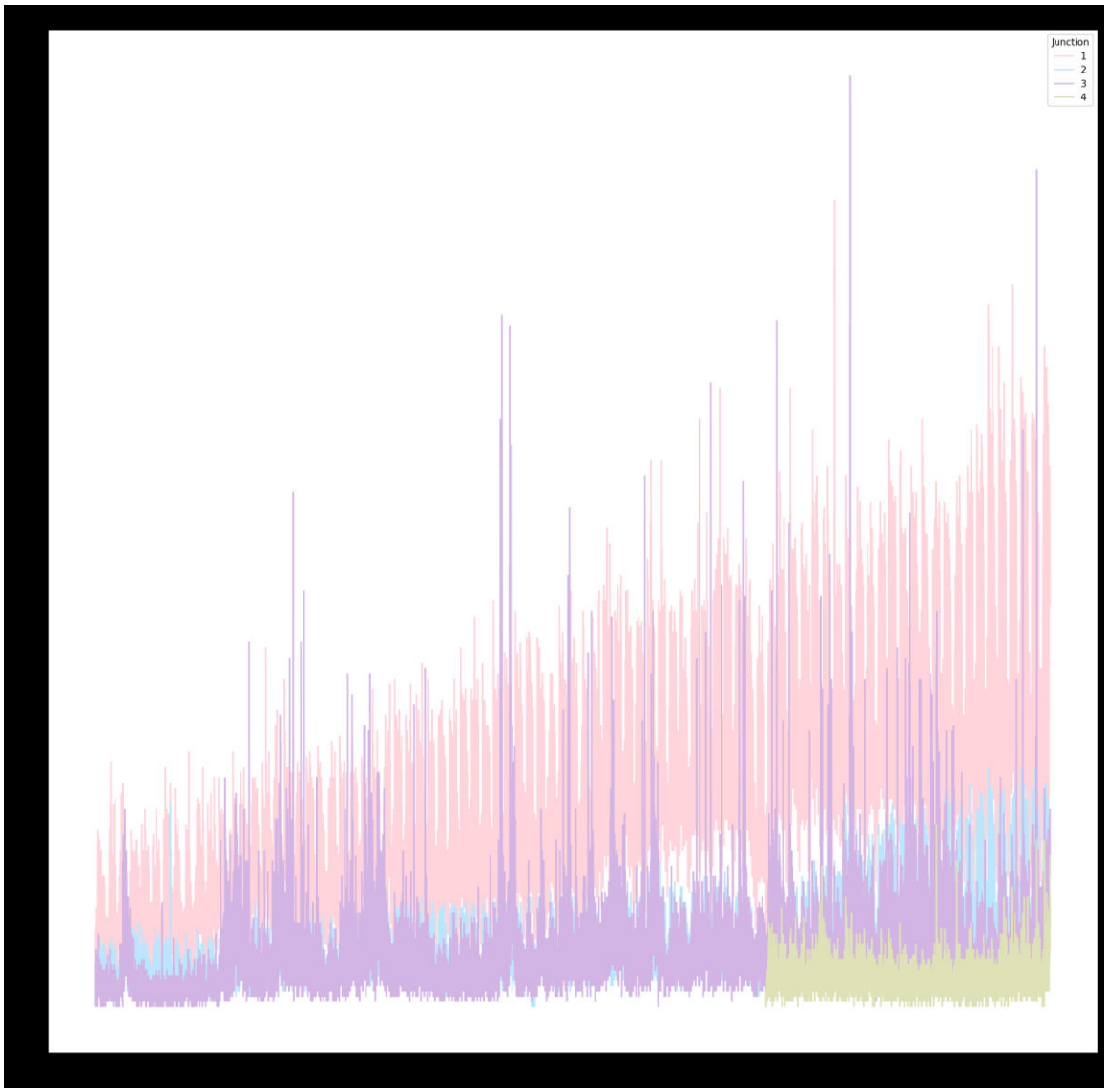
Out [2] :

	Date	Time	Junction	Vehicles	ID
0	2015-11-01	00:00:00	1	15	20151101001
1	2015-11-01	01:00:00	1	13	20151101011
2	2015-11-01	02:00:00	1	10	20151101021
3	2015-11-01	03:00:00	1	7	20151101031
4	2015-11-01	04:00:00	1	9	20151101041

In [3] :

```
palette = [ "#FFD4DB","#BBE7FE","#D3B5E5","#dfe2b6"]
plt.figure(figsize=(20, 20), dpi=150)
series = sns.lineplot(x=df_['DateTime'], y = "Vehicles", data=df_, hue="Junction", palette=palette)
series.set_title("Traffic based Junction", fontsize=18)
series.set_ylabel("Number of Veichles")
series.set_xlabel('Date')
```

Out [3] :



In [4] :

```
df_['Year'] = df_['DateTime'].dt.year
df_['Month'] = df_['DateTime'].dt.month
df_['Date_no'] = df_['DateTime'].dt.day
df_['Hour'] = df_['DateTime'].dt.hour
df_['Day'] = df_.DateTime.dt.strftime("%A")
df_.head(
```

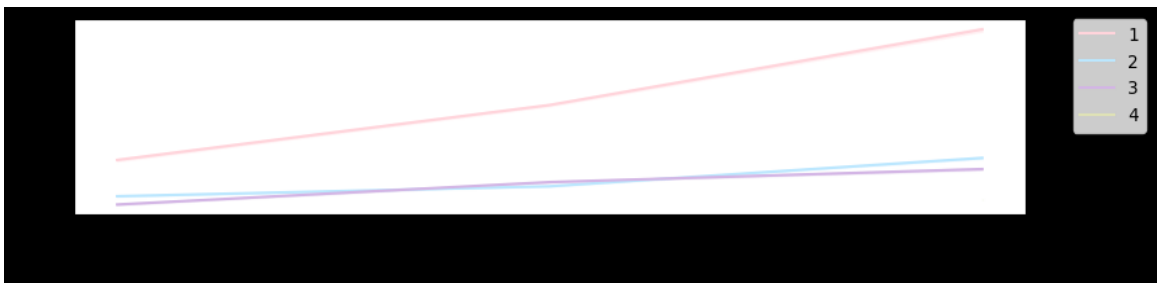
Out [4] :

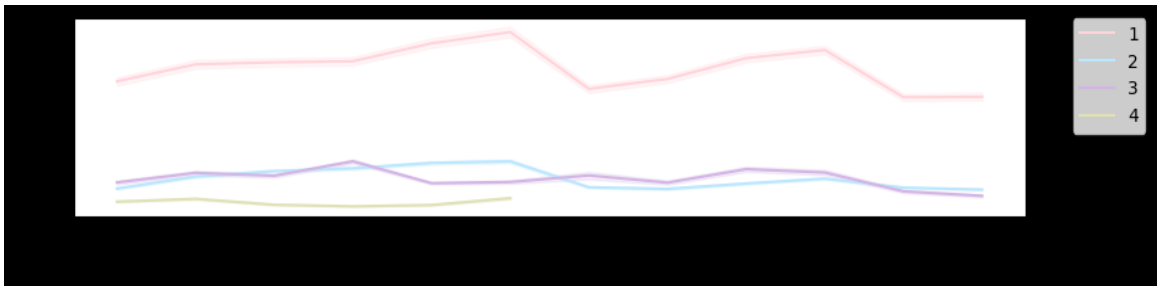
	Date	Time	Junction	Vehicles	Year	Month	Date_no	Hour	Day
0	2015-11-01	00:00:00	1	15	2015	11	1	0	Sunday
1	2015-11-01	01:00:00	1	13	2015	11	1	1	Sunday
2	2015-11-01	02:00:00	1	10	2015	11	1	2	Sunday
3	2015-11-01	03:00:00	1	7	2015	11	1	3	Sunday
4	2015-11-01	04:00:00	1	9	2015	11	1	4	Sunday

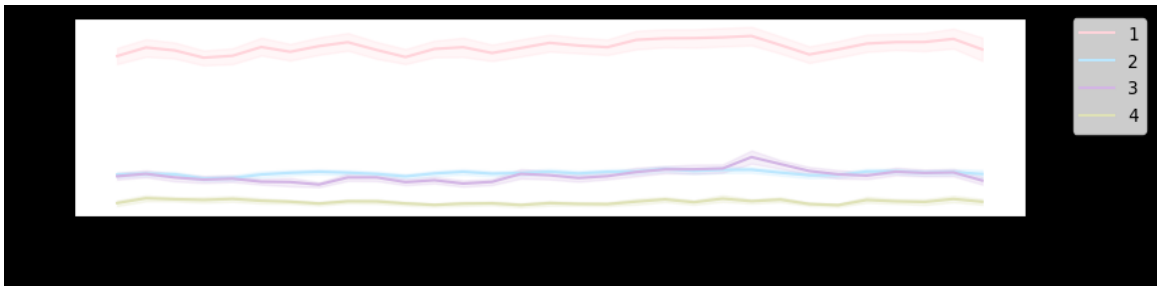
In [5] :

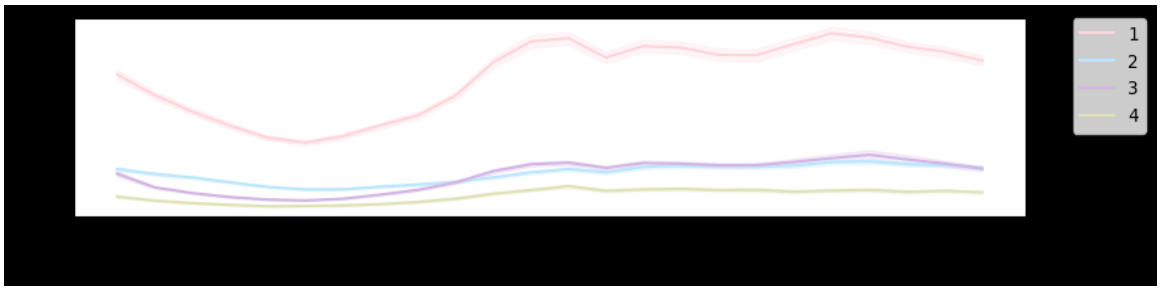
```
created_feature = ['Year', 'Month', 'Date_no', 'Hour', 'Day']
for i in created_feature:
plt.figure(figsize=(10, 2), dpi=100)
ax = sns.lineplot(x=df_[i], y='Vehicles', data=df, hue='Junction', palette = palette)
plt.legend(bbox_to_anchor = (1.05, 1), loc=2, borderaxespad=0.)
```

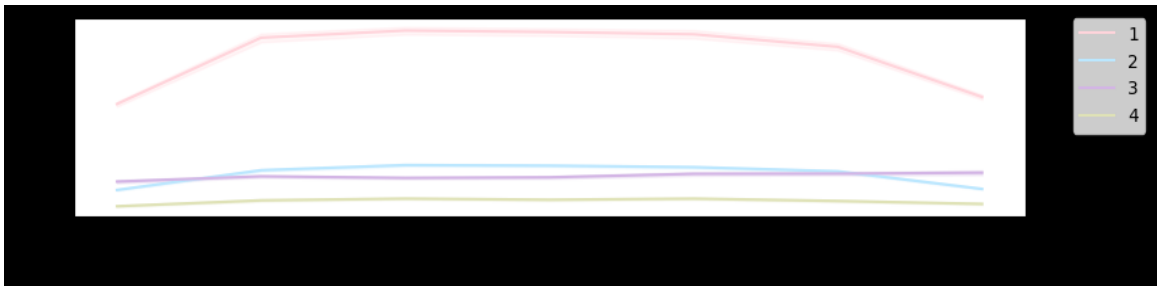
Out [5] :







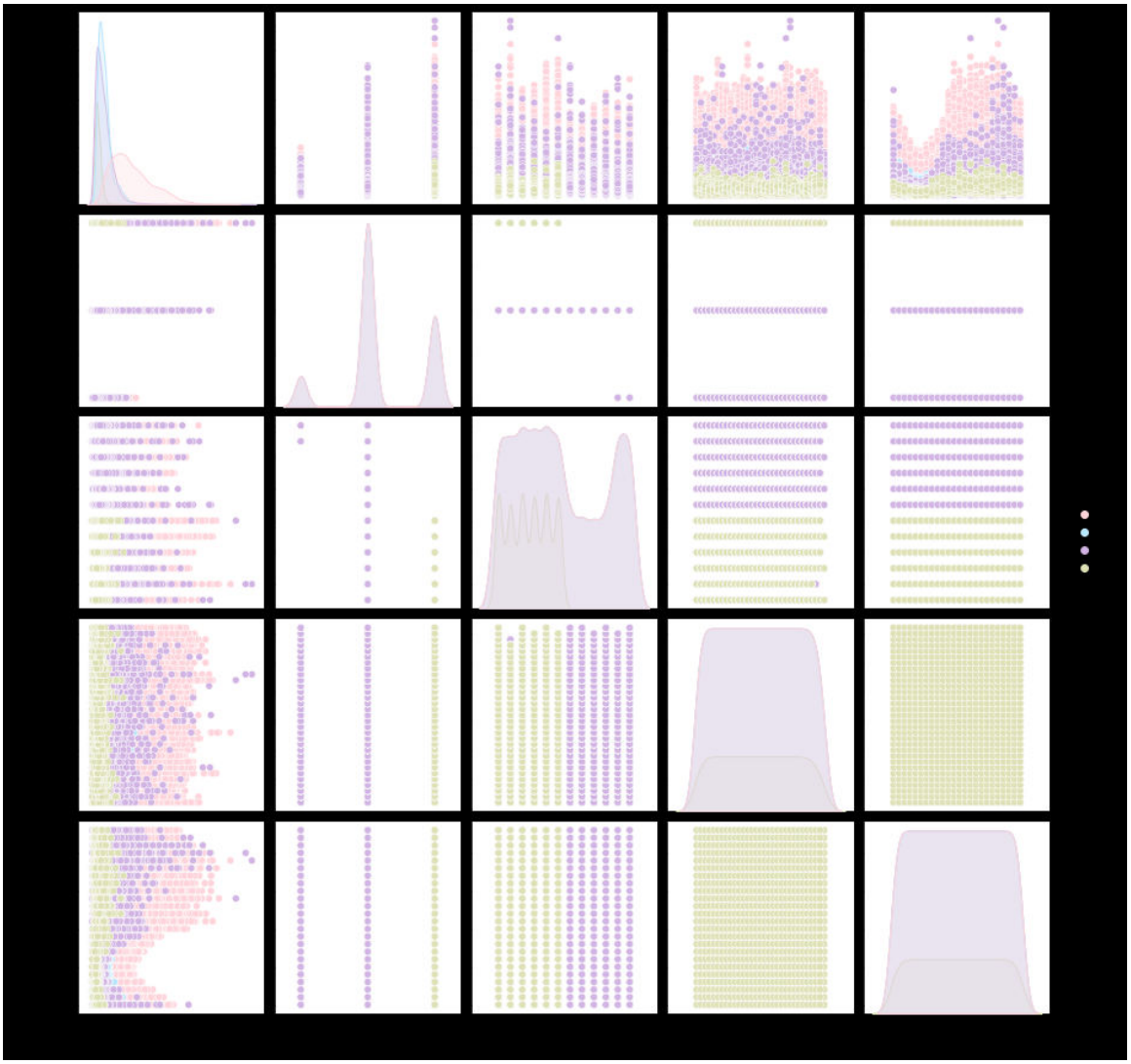




In [6] :

```
sns.pairplot(data=df_, hue= "Junction",palette=palette)
```

Out [6] :

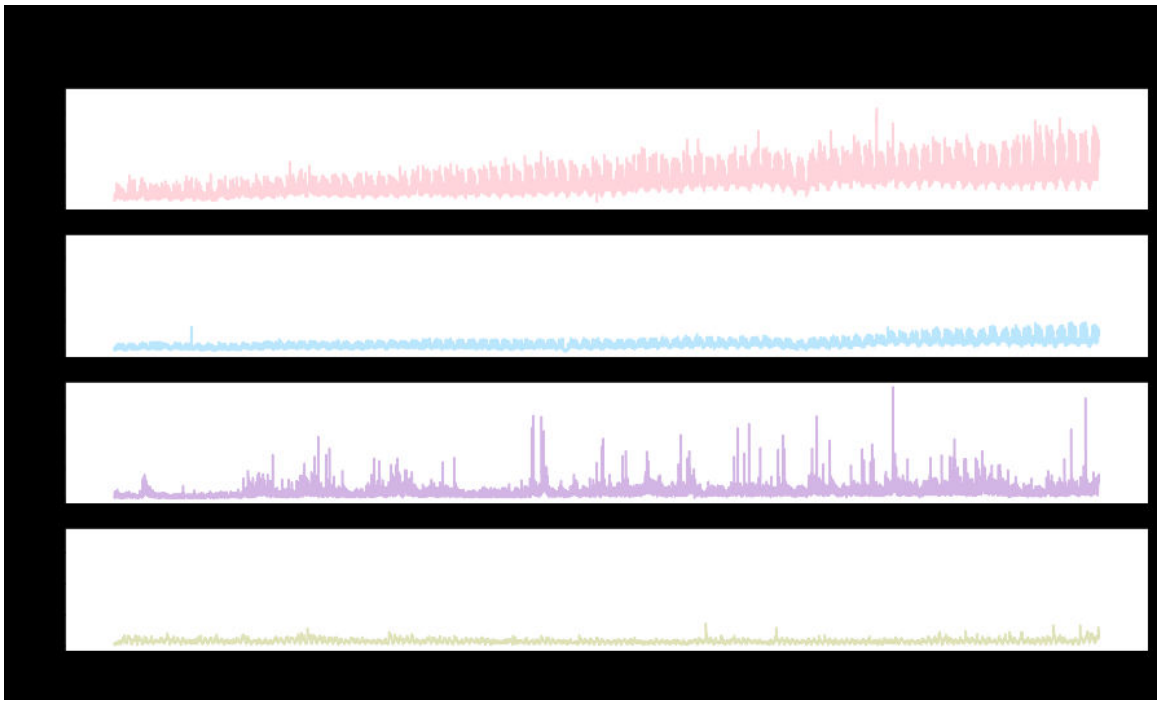


In [7] :

```
df_1 = df_junction[[('Vehicles', 1)]]

df_2 = df_junction[[('Vehicles', 2)]]

df_3 = df_junction[[('Vehicles', 3)]]

df_4 = df_junction[[('Vehicles', 4)]]

df_4 = df_4.dropna()

list_dfs = [df_1, df_2, df_3, df_4]

for i in list_dfs:

i.columns= i.columns.droplevel(level=1)
```

```
#Function to plot comparitive plots of dataframes
def Sub_Plots4(df_1, df_2,df_3,df_4,title):
fig, axes = plt.subplots(4, 1, figsize=(15, 8),sharey=True)
fig.suptitle(title)
#J1
pl_1=sns.lineplot(ax=axes[0],data=df_1,color=palette[0])
#pl_1=plt.ylabel()
axes[0].set(ylabel ="Junction 1")
#J2
pl_2=sns.lineplot(ax=axes[1],data=df_2,color=palette[1])
axes[1].set(ylabel ="Junction 2")
#J3
pl_3=sns.lineplot(ax=axes[2],data=df_3,color=palette[2])
axes[2].set(ylabel ="Junction 3")
#J4
pl_4=sns.lineplot(ax=axes[3],data=df_4,color=palette[3])
axes[3].set(ylabel ="Junction 4")
```

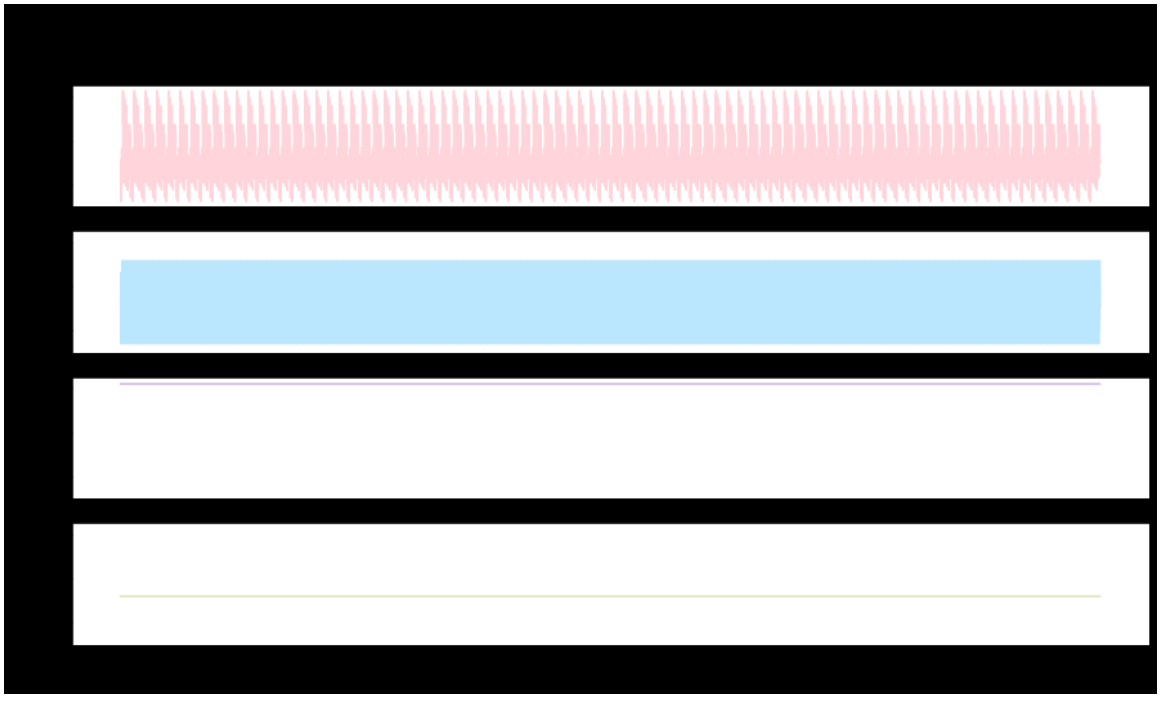
Out [7] :



In [8] :

```
Sub_Plots4(df_N1.Diff, df_N2.Diff,df_N3.Diff,df_N4.Diff,"Dataframes After        Transformation")
```

Out [8] :



Conclusion:

❖ In the quest to build a house price prediction model, we have embarked on a critical journey that begins with loading and preprocessing the dataset.We have traversed through essential steps, starting with importing the necessary libraries to facilitate data manipulation and analysis.

❖ Understanding the data's structure, characteristics, and any potential issues through exploratory data analysis (EDA) is essential for informed decision-making.

❖ Data preprocessing emerged as a pivotal aspect of this process. It involves cleaning, transforming, and refining the dataset to ensure that it aligns with the requirements of machine learning algorithms.

❖ With these foundational steps completed, our dataset is now primed for the subsequent stages of building and training a house price prediction model.