A Machine Learning Approach to Predicting and Mitigating Traffic Congestion

by

Abu Fatah Mohammed Faisal
20301240
Chowdhury Zaber Bin Zahid
20301256
Walid Ibne Hasan
20301103
Shuvo Talukder
23141068

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
March 2024

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Faisal

Abu Fatah Mohammed Faisal
20301240

Zaber

Chowdhury Zaber Bin Zahid
20301256

Walid

Walid Ibne Hasan
20301103

Shuvo

Shuvo Talukder
23141068

Approval

The thesis titled "A Machine Learning Approach to Predicting and Mitigating Traffic Congestion" submitted by

1. Abu Fatah Mohammed Faisal (20301240)
2. Chowdhury Zaber Bin Zahid (20301256)
3. Walid Ibne Hasan (20301103)
4. Shuvo Talukder (23141068)

Of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on March, 2024.

Examining Committee:

Supervisor:
(Member)

A.M. Esfar-E-Alam
Senior Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Traffic congestion has notable effects on urban mobility, impacting thousands of people on a daily basis which hampers economic productivity and environmental sustainability. This research represents an extensive approach to address the multi-faceted affairs of traffic jams through data analysis, machine learning modeling and prediction analysis. This research emphasizes four key dimensions. Such as traffic patterns, data preprocessing, model implementation and result analysis.

This research starts by diving deep into the complex dynamics of the urban traffic jam, recognizing the crucial challenges such as outdated infrastructure, suboptimal traffic signal synchronization, and the unstable navigation system exemplified by Google Maps. Through diligent data exploration, data preprocessing, temporal features which we fetched from the dataset which enable a deeper understanding of traffic congestion patterns and temporal dependencies.

By developing a robust machine learning model, leveraging the Random Forest Regressor, we have predicted the number of vehicles across four junctions. The Model class framework summarizes different preprocessing steps, model training, evaluation metric calculation and prediction abilities. The prediction capabilities of the model extend to forecasting future traffic volumes for the coming four months which empowers the stakeholders with proactive decision-making insights. Among the key takeaways that we can have from the research are the model's versatility, adaptability to different traffic prediction scenarios, and its ability to capture temporal patterns and predict future outcomes.

To conclude, the research presents a holistic framework for better comprehension, forecasting and optimization of the traffic patterns with effects which extend to the urban planning, infrastructure management and traffic management strategies.

Acknowledgement

The completion of this research paper has been a collaborative effort, and it is with immense gratitude that we express our appreciation to all those who have contributed to its fruition.

We extend my deepest thanks to our supervisor, A.M. Esfar-E-Alam, for his invaluable guidance, unwavering support, and insightful feedback throughout the entire research process. His expertise and commitment played a pivotal role in shaping the direction of this study and elevating its quality.

We also express gratitude to our families and friends for their unwavering support, understanding, and encouragement during the ups and downs of the research journey. Their patience and belief in the significance of this work have been a driving force. This research paper stands as a testament to the collaborative spirit and collective efforts of all those mentioned above. While any errors or omissions remain my responsibility, it is their collective support that has made this endeavor possible.

Table of Contents

Chapter 1

Introduction

Traffic congestion is still a big problem for transport systems that need to work well, even though the ways people move around cities are always changing. It's becoming more and more important to come up with good ways to predict and get rid of traffic jams as cities grow and more people drive cars. The main focus of this thesis is on using advanced machine learning methods to predict when traffic will be heavy on a certain road.

Traffic delays have many aspects of it. It is not just a hassle for society but its effects go far beyond that. It hampers the economy of a country, the environment, health, and so many other things. Congestion of traffic has so many effects that this study has tried to come up with a new, smart, and logical solution using machine learning.

The approach way is meant to simplify traffic prediction in a much smarter way. Machine learning has the potential to find complex trends and changing connections in large amounts of data. This could help us go beyond the limits of current traffic control methods.

Before getting into the specifics of the suggested answer and what benefits it might have, it is important to understand how serious the traffic problems are. Traffic jams are not only irritating every day, but they also have a real economic cost because they cause people to lose time at work and use more petrol. Also, traffic delay is bad for the earth because it leads to more pollution and worse air quality. Unpredictable traffic conditions cause worry that lowers people's quality of life, which in turn lowers their mental health and general happiness.

As traffic jams are so complicated, they need to be predicted and managed in new and subtle ways. Advanced machine learning methods are being looked into because traditional models often fail to capture the complex dynamics of traffic trends. The goal of this study is to help come up with methods for traffic control that go beyond simply predicting what will happen and include all the different aspects of traffic problems.

The study doesn't use the usual ways of managing traffic. Instead, it uses advanced machine learning methods to identify and deal with traffic jams before they happen. By finding complicated patterns and connections in datasets, these methods have shown promise in a number of areas. We want to get around the problems with current models of traffic forecasts by using machine learning to make the process more accurate and proactive.

Advanced machine learning methods are chosen because they can handle links that don't follow a straight line and exchanges that are very complicated in the data. Because traffic trends change all the time and are affected by many linked factors, we need a model that can accurately reflect these subtleties. Machine learning is being used in this study to try to go beyond the limits of standard traffic control and forecasts.

The most important part of this study is not only figuring out how traffic will behave, but also pushing for a shift in the way traffic is managed. The goal is to give transport and urban planners the tools they need to take preventative steps. Advanced machine learning techniques provide a foundation for devising strategies to mitigate congestion before it reaches critical levels, thereby optimizing traffic flow.

Adding traffic forecasts based on machine learning fits with the bigger idea of "smart cities," which are places where data and technology come together to make services more efficient and last longer. By effectively predicting and controlling traffic, cities can make the best use of their resources, cut costs, and improve movement in cities as a whole. The suggested answer goes beyond the current issue of traffic jams and helps create a future where towns are able to change and do well as the world becomes more urbanised.

As with any new idea, there are problems to deal with, and this study recognises the need to do so in a smart way. In the real world, unpredictable events often happen that can throw off traffic trends. It is very important to make sure that the machine learning methods that are picked can be used to solve these problems. The study focuses on improving and adapting the model all the time to make it better at dealing with problems and make sure it works well in all kinds of traffic situations.

Thinking about the benefits of this study goes beyond just easing traffic right now. Management of traffic trends that is planned ahead of time affects many areas, making society better as a whole. Simplifying transportation can help businesses by cutting costs and making them more efficient overall. Residents' daily lives are clearly better because easier traffic flow means they spend less time travelling and live in a more regular and stress-free city.

With the help of advanced machine learning methods, this paper is a big step towards solving the problems caused by traffic jams. Focusing on a specific road and using these methods shows a dedication to learning about the complexities of traffic trends and making an answer that is more complex and useful. Adding machine learning models to traffic predictions is at the heart of changing how people move around in cities as they try to become better and more environmentally friendly. Using advanced machine learning methods, this study aims to not only correctly predict traffic conditions but also pave the way for a future where towns can handle and improve their transport systems on their own. When you combine data-driven decision-making with real-time prediction analytics, you get better, safer, and more environmentally friendly ways to get around cities. When we accept this paradigm shift, advanced machine learning techniques shine like a light, guiding us through the complicated world of traffic and helping towns thrive in a future where urbanisation is changing the landscape.

1.1 Problem Statement

In today's urban world traffic jams is a common problem. It's creating a fabric of metropolitan life, affecting millions daily, and casting a shadow on economic productivity and environmental sustainability. As cities people are getting increased so is the necessity of vehicles. As a result, it creates a problem that necessitates a comprehensive exploration. This is affecting traffic jam issues in three ways such as unreliable Google Maps, limitation of proper research papers, and escalating concerns surrounding the reliability of navigation systems.

1.1.1 Traffic Jam Issue

Transportation system is often seen in cities or urban regions. But because of traffic jams, it creates various problems, such as wasting time, frustration and so more. Furthermore, it indicates our mistake in the traffic system for we face this problem in our daily life.

<u>Outdoor Infrastructure:</u> Planning for building roads was made during the era when vehicles were not that available. As a result, it was planned for a few cars but since now vehicles have been increasing roads are becoming busy and hard to maintain traffic.

<u>Suboptimal Traffic Signal Synchronization:</u> Another major reason is traffic lights are automated. They changed based on the time we have programmed on it. As a result, traffic is not well maintained and create slow movement and greed lock in some junctions.

<u>Lack of real-time Time Adaptive strategies:</u> Since traffic systems are maintained at a fixed rate of time real scenario-based traffic is not maintained. As a result, they aren't very good at fixing traffic jams or making commuting easier for people.

1.1.2 Unreliability of Google Maps

Modern tools like google Maps are used to make traffic much easier but still, it has its limitations as a result sometimes people can not fully rely on it for getting around their daily routine.

<u>Inaccuracies in Real-Time Traffic Information:</u> Google map uses the latest data to give users proper routes to the way point. But sometimes the route is not updated. As a result, wrong info gives users a negative expression to use it for their daily to daily life journey.

Suboptimal Route Suggestions: Navigation systems use algorithms to determine the optimal routes for users. Despite significant advancements in technology, these algorithms still have limitations and may lead users to take longer or more crowded routes due to over-reliance on historical data or algorithmic flaws.

Inability to Adapt to Dynamic Traffic Conditions: Since urban areas' traffic is changing continuously google map needs to update its data also continues. But it does not immediately show the new route to the user when something huge thing changes suddenly. This is something google map is lacks.

1.1.3 Limitations of the Thesis Paper

As this thesis wants to improve traffic systems by using machine learning we need to take ideas and hints from previous research done on this topic.

The complexity of Urban Traffic Dynamic: As the city's traffic is complex, it is happening day by day as a result finding a solution for this complicated thing is very hard. As each urban city has its problem we have to come up with the right solution for them.

Reliance on External Data Sources: Having the right data, like current traffic information and good feedback from users the success is dependent. The solution might be effected as sometimes these data sources may not be accurate or perfect. Since we have to work based on this things might get less predictable.

Potential Technological Constraints: Things like having the right introspection, not enough money, or cities not being ready for the new system might get tricky even if we use technology. How well the solution can be used widely and how practical they are can be affected by this kind of limitation. Therefore it is very important to understand which tech is available and how to fit it into the work. Taking note of these difficulties, the thesis aims to strike a balance between theory and application, providing the framework for further studies that will focus on certain aspects of traffic improvement.

1.1.4 Navigating the Intricacies of Urban Mobility Transformation

To conclude, as seen prominently with Google Maps, emphasizes the urgent requirement for creative solutions in urban mobility the recognition of traffic jams, and the inconsistency of navigation systems. All contribute to congestion in cities, emphasizing the need for significant systemic improvements the convergence of obsolete infrastructure, ineffective traffic signal coordination, and a lack of real-time adaptive approaches.

At the same time, another level of complexity to commuting is the increasing dependence on navigation systems. For city dwellers issues like inaccurate real-time traffic updates, less-thanideal route recommendations, and the inability to adjust to changing traffic situations make the daily commute even more challenging. A reassessment of current navigation approaches is called for as a result.

Yet, the path to revolutionizing urban mobility encounters its hurdles. The parameters within which this thesis functions are the intricate nature of urban traffic patterns, dependence on external data, and possible technological limitations. It highlights the necessity for a nuanced grasp of the existing challenges acknowledging these inherent limitations doesn't diminish the importance of the proposed research. In urban settings, this sets the stage for future efforts aimed at streamlining traffic and improving navigation reliability.

This research aims to make a significant contribution to the dialogue surrounding traffic congestion, navigating the complexities of urban mobility evolution. It endeavors to provide valuable guidance, by offering insights that bridge theory and practice. The discoveries from this research could serve as a guiding light for policymakers, urban planners, and technology innovators, as urban environments undergo continual transformation. In the end, they might get us closer to a time when traffic jams are lessened and navigation systems are trusted partners in the pursuit of sustainable urban transportation.

1.2 Research Objective

The study aims to create a strong and accurate model for predicting how traffic will flow at different intersections. The main goals are the following:

Data Exploration and Understanding:
Look at the given traffic information and figure out what it is made of, such as statistical reports, distribution patterns, and time trends. Also, look into how things like time of day, day of the week, and the position of the intersection affect the amount of traffic.

Data Preprocessing:
Take useful timing features (Year, Month, Day, Hour) out of the timestamp data to make training the model easier. Also, look into whether you need to get rid of some sections, like the "ID" column, to make the model work better.

Visualization and Pattern Analysis:
Make visualizations, such as histograms and time series plots, to learn more about how and where vehicles are travelling at different intersections. Also, look at how the factors are related to each other and how traffic moves at each point during different times.

Normalization Techniques:
Use Z-score normalisation on the dataset and check how it changes the distribution and association of traffic data. Then, check how well normalisation works to make the model better at finding patterns in how traffic moves.

Outlier Detection and Handling:
Box plots can help you find outliers in the flow data. Also, we have to find ways to

deal with outliers to make the model more reliable.

Time Series Analysis:

We have to use autocorrelation analysis to understand how the travel data has changed over time. Furthermore, we have to look at how non-identical time delays are related and whether they can be used to predict future traffic numbers.

Modeling Approaches:

We have used machine learning models, like the Random Forest Regressor, to predict the number of traffic. Then, we trained our models on the original dataset and the normalized dataset to see the performance of the model. The lag features can also be added to the models to see how they handle time dependencies in the data.

Model Evaluation:

For the evaluation appropriate metrics should be used to assess the performance of the models. R^2 score and Root Mean Squared Error (RMSE) are used to compare the performance of the models trained on the different datasets (original, normalized, and lag features).

Feature Importance Analysis:

“Random Forest Regressor” is used to see the importance of the features for the prediction of traffic numbers. A picture of the main traits can help the model make predictions for each point.

Predictive Modeling for Future Traffic:

Use dated data and machine learning models to come up with a way to guess how much traffic will be in the next four months. Lastly, check how accurate and dependable the forecasts are for each junction.

Chapter 2

Literature Review

Traffic congestion in urban areas negatively impacts people's quality of life, air quality, and transit efficiency. Researchers explored many strategies to improve traffic flow. This part gives a detailed overview of the literature in this topic, emphasizing key concepts and techniques.

Traffic lights play an important role in developing smart cities by reducing traffic congestion and pollution in urban areas. A bi-level optimization approach can help address the optimal traffic signal configuration problem [7]. There is also suggested work, including a complete model for anticipating traffic flow and an algorithm for optimizing lane allocation at an Austrian toll plaza [11]. The time series-based traffic forecasting method has a long-term prediction error of less than 15%, and the optimization technique, which uses a camera-based monitoring system, effectively decreases travel times by up to 6% and queue lengths by up to 30% [11].

A study focused on the often-overlooked element of cars redirecting in response to changes in signal timing in synchronized traffic signal control. Using a Genetic Algorithm (GA) and a network equilibrium model, it will increase the timing of signals efficiency, especially in busy networks [2].

A unique technique to reducing traffic congestion employs reinforcement learning, with traffic flow optimization implemented as a Markov Decision Process [5]. The simulation experiments show that using Q-learning and traffic forecasts results in dramatically reduced traffic [5].

There are certain limitations and possibilities in using GPS data to analyze the flow of traffic behavior. Though GPS provides useful space-time details for spotting congestion, its limited resolution and absence of lane-specific data make it difficult to evaluate traffic conditions comprehensively [4].

A research project proposed a stochastic model for major road junctions that uses traffic-responsive signalization algorithms based on cut-off queue length and density [1]. The study seeks to reduce total time by providing information about optimizing traffic flow at single crossings using a probabilistic cellular automata framework [1].

There is a study that focuses on optimizing traffic flow at urban junctions by tackling the issues provided by rising traffic and the limits of existing traffic light systems. Using genetic algorithms, the study investigates the impact of queue length, green time, and cycle time giving a new approach to self-tuning traffic management [3].

Ali and others perform a thorough investigation of Dhaka's serious traffic problem, blaming it on population expansion, poor infrastructure, and socioeconomic factors that drive private automobile ownership [16].

A study investigates the use of machine learning in traffic management, tackling the complications caused by increased worldwide traffic. The study highlights the ability of machine learning algorithms to improve precision as well as effectiveness while managing traffic data [15].

There is an experiment that shows how machine learning models, such as long short-term memory and frequent networks, can forecast traffic on the road. Using data from a well-known traffic simulator, the work demonstrates real-time applicability, especially for evaluating variable phrase road sign speeds, and provides a possible alternative to classic microscopic traffic models [19].

An in-depth examination of the rapidly increasing topic of traffic congestion forecasting, with a focus on the role of artificial intelligence and machine learning algorithms. It divides numerous AI approaches, concentrating on data from the past and the present, and provides a systematic review of strengths and shortcomings in the current research environment [12].

Chhatpar and colleagues address the rising traffic difficulties in Indian cities by presenting a machine learning approach for predictive analysis of traffic based on the Back Propagation Neural Network [8]. Their Android software uses live traffic information to provide offline predictions and route recommendations to reduce congestion and improve efficiency, with an emphasis on reducing smartphone power use [8].

A research project presents eRCNN, a deep learning solution for continuous traffic speed forecasting, which uses spatiotemporal information and error feedback neurons to increase accuracy during sudden occurrences [6]. Testing on Beijing's ring roads indicate the model's high predictive capacity when compared to cutting-edge approaches, highlighting its potential for real-world traffic speed prediction and congestion source recognition [6].

Kaushik discusses the significant issue of traffic in Ad Hoc mobile networking and road traffic systems, focusing on the changing concept of vehicles equipped with advanced sensors and communication tools [18]. He investigates the possibilities of Mobile Ad Hoc Networks (MANETs) and Vehicle Ad Hoc Networks (VANETs) in Intelligent Transportation Systems by combining massive amounts of data, sensors, and machine learning to predict and manage traffic congestion [18].

AA research study offers MSR2C-ABPNN, a Smart Road Traffic Congestion Control system that uses Artificial Neural Networks (ANN) [9]. The system they develop uses backpropagation for training and tries to forecast and reduce traffic congestion, improving transparency and efficiency in urban traffic networks. The study emphasizes the value of machine learning approaches in delivering intelligent transportation solutions for metropolitan populations [9].

A study conducted a complete assessment of target recognition in smart cities, concentrating on traffic congestion utilizing deep learning-enabled UAVs [17]. The review focuses on advances in deep learning algorithms, highlighting accuracy enhancement, computational efficiency, and future research directions for optimizing target detection in UAV photography [17].

A research project investigates the problem of elephant flows generating network congestion in multimedia applications. The research addresses constructing traffic forecasting models using deep learning algorithms such as H2O, Deep Autoencoder, XGBoost, GBM, and GDF, that have excellent validation accuracy levels [20].

A paper describes a Smart Traffic Management Platform (STMP) that uses unsupervised online incremental machine learning, deep learning, and deep reinforcement learning [10]. Recognizing the limits of conventional AI methodologies, the platform combines incompatible big data sources from IoT, smart sensors, and social media to improve traffic management as demonstrated on a large dataset from Victoria, Australia [10].

There is a study that thoroughly examines the optimization of Random Access Channel (RACH) processes in wireless communications, contrasting machine learning (ML) and non-ML methodologies [13]. The proposed decoupling learning strategy (DLS) is notable for its effectiveness, as it uses supervised learning for traffic prediction and provides a diverse method to access control optimization [13].

A study on urban traffic congestion investigates the usefulness of machine-learning and deep-learning algorithms for predicting traffic flow at intersections [14]. It also demonstrates that the Multilayer Perceptron Neural Network (MLP-NN) beats other approaches in terms of predicted accuracy and training effectiveness, making it a suitable candidate for use in smart traffic light controls [14].

Chapter 3

Methodology

In this section, we thoroughly discussed our machine learning model that we have used here and how we implement it step by step. Moreover, we discussed and explored our dataset with different approaches and graphs and we also explained how we worked on our dataset in order to implement our machine learning model with important features for achieving the results.

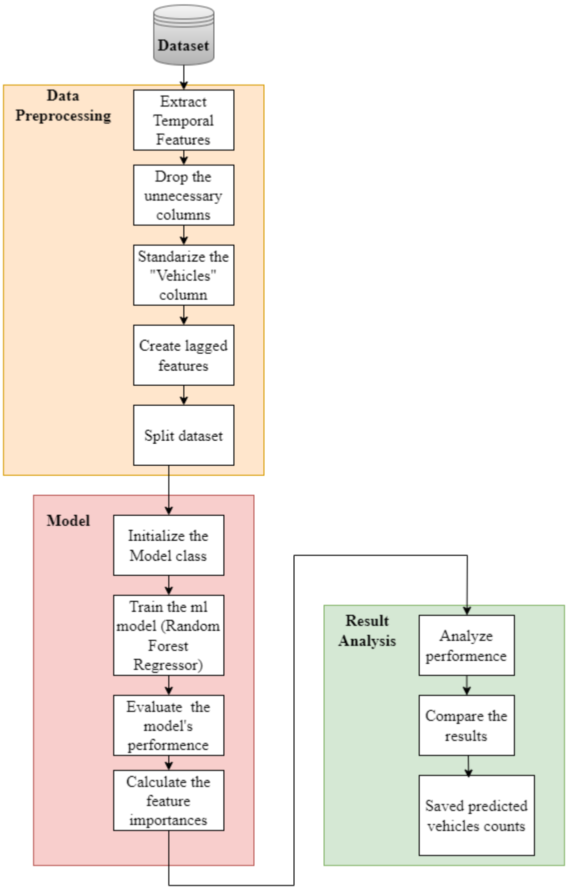

Dataset
Data Preprocessing
Extract Temporal Features
Drop the unnecessary columns
Standarize the "Vehicles" column
Create lagged features
Split dataset
Model
Initialize the Model class
Train the ml model (Random Forest Regressor)
Evaluate the model's performence
Calculate the feature importances
Result Analysis
Analyze performence
Compare the results
Saved predicted vehicles counts

Figure 3.1: Flow Chart of the Proposed Model

3.1 Description of The Model

To predict the traffic condition by using the counts of vehicles over four junctions, we used the “Random Forest Regressor” model as we want to predict the continuous outcome of traffic jams. To use this model, we need to modify the data so that we can get the best results from the prediction, creating a class frame for the model so that we can represent the model’s results and get to know the comparison between the junctions etc. So, we proposed a model of our work plan so that we could predict the traffic jams at those junctions.

To begin with, we need to load the dataset. Our dataset has four columns. These columns are: DateTime, Junction, Vehicles, ID. Here, the ID column is not important for our model. So, we will work on the other three columns in order to run our selected model. The junction column has four junctions and each junction has multiple vehicles at different times.

Then, we plotted some graphs for the purpose of data analysis using the data we have. This data analysis will help us to visualize the significance of the data for running the model. Also, these analyses will help us to understand the correlation between the data.

On the other hand, we needed to do some data preprocessing so that we can get the best outcome for the prediction. In this part, we worked on the dataset carefully. We also made some separate datasets for comparing the results so that we can verdict the final result from our model.

However, we needed to make metrics for running the multiple models for each junction as we mentioned earlier that we need to make comparisons of our results for each junction so that we can predict the traffic jam with best strategies. After that, we make a class frame for running our selected model. It specifies a class in Python named Model, which is meant to be a general foundation for developing and assessing machine learning models. It trained and tested the model on the dataset before returning the results with our chosen machine learning model. We also used additional techniques in this class, such as the precondition method for feature selection, the fit method for fitting the model to training data and forecasting the results, and so on. In addition, two functions are available for calculating the R^2 Score and Root Mean Squared Error (RMSE) values. These approaches are invoked sequentially and essentially carry out all of the procedures required to train and evaluate the model. To estimate traffic patterns this course offers a complete framework for training, assessing, and visualizing machine learning models that are specifically tailored. In time series data as well as the impact of numerous variables on vehicular traffic by utilizing a number of methodologies and capabilities the model is designed to capture the complexities of temporal dependencies. In order to concentrate more on the methods:

Initialization:
An initialization function that accepts crucial parameters are started its trip with the Model class:

name: For easy identification and reference a unique identity for the model is name.
data: A set of data contains past traffic information that was parsed with pandas and indexed using DateTime.
predict_features: The value that is used to predict (in this example, the number of vehicles).
test_size: The ratio of the dataset dedicated for testing, which allows for robust model evaluations.
ml_model: The machine learning model used for the task. In this case, the Random Forest Regressor is used because of its adaptability and performance.

Data Preprocessing and Feature Extraction:

To prepare for effective model training, the model performs thorough data preparation. Temporal features such as Year, Month, Day, and Hour were taken from the Date Time index, which improves the model's capacity to detect trends across periods of time. The initial method prepares the features and target variables, while the information is then split into training and testing sets using scikit-learn's commonly used train_test_split function.

Model Training and Evaluation:

The Model class's fundamental capability is being able to train and assess machine learning models. The fit approach manages the training process, using the specified machine learning model to discover patterns from the training data. Afterwards, the model's prediction abilities are tested on the reserved test set. The cal_rmse and cal_r2_score algorithms are used to determine performance measures such as RMSE and R^2.

Visualization for Exploration:

Analyzing the data is an important stage in model creation. The Model class has utilities for constructing informative visualizations. The make_hist function produces histograms with kernel density estimate (KDE) plots, which give information on the pattern of vehicle counts at various junctions. Time series plots (make_time_series_plot) offer a dynamic depiction of vehicle counts across time, allowing for a better understanding of patterns over time at every junction.

Normalization Techniques and Outlier Handling:

Acknowledging the necessity of data normalization, the model includes Z-score normalization. The vehicle counts, resulting in a mean of 0 and a standard deviation of 1 are function is modified by the standardization lambda. The impact of normalization are illustrate by Histograms.

Furthermore, the presence of outliers and identify potential anomalies in the data are investigated by boxplots which the model use. In order to improve the model's resilience the research objective includes ways for dealing with outliers.

Time Series Analysis and Autocorrelation:

The model does time series analysis with the temporal nature of traffic data . The time dependent nature of the data are provided information by autocorrelation and partial autocorrelation charts. How previous observations influence future traffic patterns are determine by analysis.

Feature Importance and Lagged Data:
An important aspect of model interpretability is feature significance. It usess a bar plot to illustrate the top features that contribute to the model's predictions. Furthermore, Resulting in a delayed dataset that improves the model's knowledge of time series patterns, the model uses lagged data to capture temporal dependencies.

Predictions for Future Traffic:
This method doesn't just look at past data. It also uses smart predictions to guess how much traffic there will be in the future. The Model class does this by looking at old data and using machine learning to make guesses about how many cars will be on the road in the next four months. This smart approach doesn't just show off how good the model is at guessing. It also gives us helpful hints about what might happen in the future. That means we can plan better and decide where to put our resources, like building roads or managing traffic, in a smarter way.

Consequently, we can say that the Model class is a flexible and all-inclusive framework made to manage the intricacies of traffic forecasting. The model provides a complete approach to assessing and forecasting vehicular traffic trends, ranging from rigorous data pretreatment and standardization to advanced time series analysis and feature importance display. Because of its adaptability, researchers and practitioners can quickly integrate it with a range of machine learning models and customize it to fit specific datasets and forecasting requirements. The model is prepared to provide significant insights into the subject of traffic prediction because it blends robust methodologies with captivating graphics.

3.2 Description of the Data

The dataset we have used emphasizes predicting traffic on four distinct junctions. Various visualization techniques and statistical methods have helped us to get a comprehensive description and analysis of the data.

Data Overview:
The dataset that we have used in the research is taken from Kaggle, a well-known site for the datasets. The dataset given in this link is an important resource for traffic prediction analysis and studies. It provides substantial data points that are needed to understand traffic patterns and predict future trends. Kaggle has allowed us to have access to high-quality data for robust research and meaningful conclusions.

The dataset shows the traffic levels at different junctions. The dataset for each entry, stores the date and time of observation, the number of vehicles, and the specific junction number.

DateTime	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

Table 3.1: The first five rows of the dataset

	Junction	Vehicles	ID
count	48120.000000	48120.000000	4.812000e+04
mean	2.180549	22.791334	2.016330e+10
std	0.966955	20.750063	5.944854e+06
min	1.000000	1.000000	2.015110e+10
25%	1.000000	9.000000	2.016042e+10
50%	2.000000	15.000000	2.016093e+10
75%	3.000000	29.000000	2.017023e+10
max	4.000000	180.000000	2.017063e+10

Table 3.2: Some description of the data in the DataFrame

Examining statistical summaries, plotting histograms, time series analysis, and correlation matrices are included in the initial exploration.

Data Exploration:
Extracting temporal components such as year, month, day, and hour from the timestamp to facilitate deeper analysis is involved in initial data.

Line plots showcase variations in traffic volume over the years, months, and even finer time intervals like days and half-days.

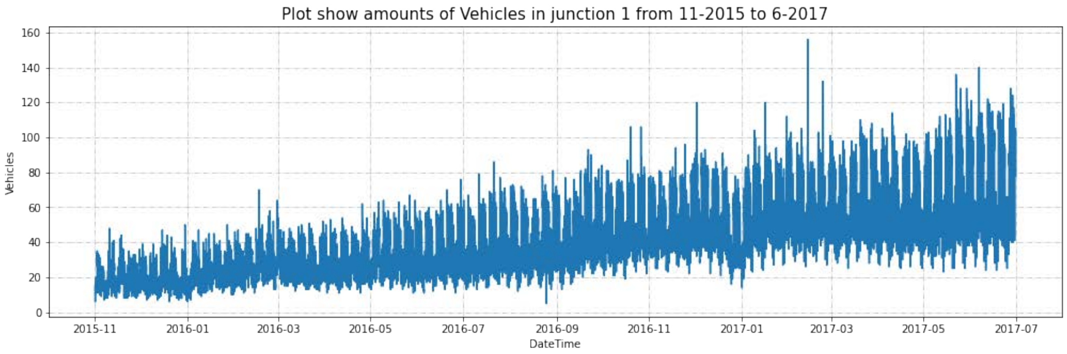


Figure 3.2: The amounts of vehicles in junction 1

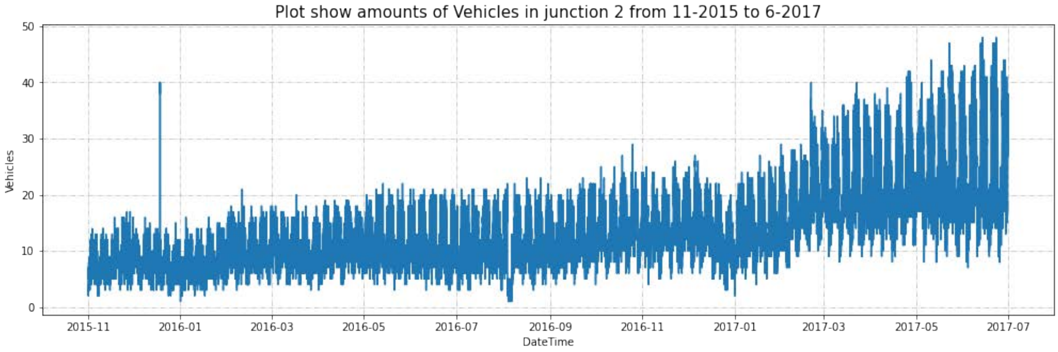


Figure 3.3: The amounts of vehicles in junction 2

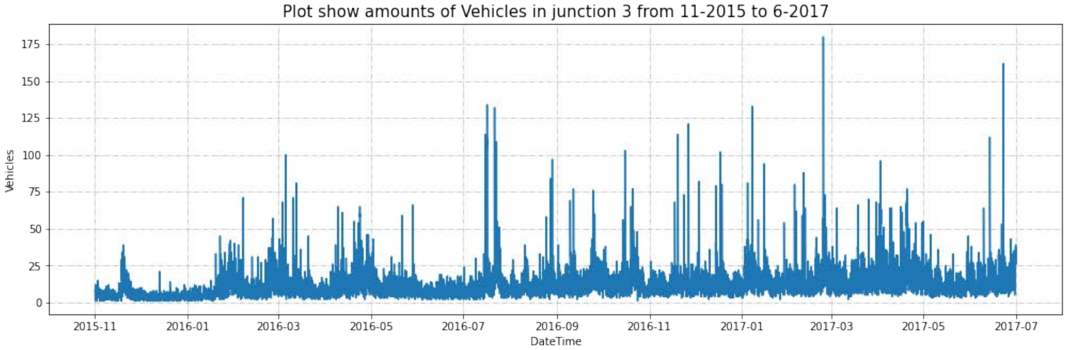


Figure 3.4: The amounts of vehicles in junction 3

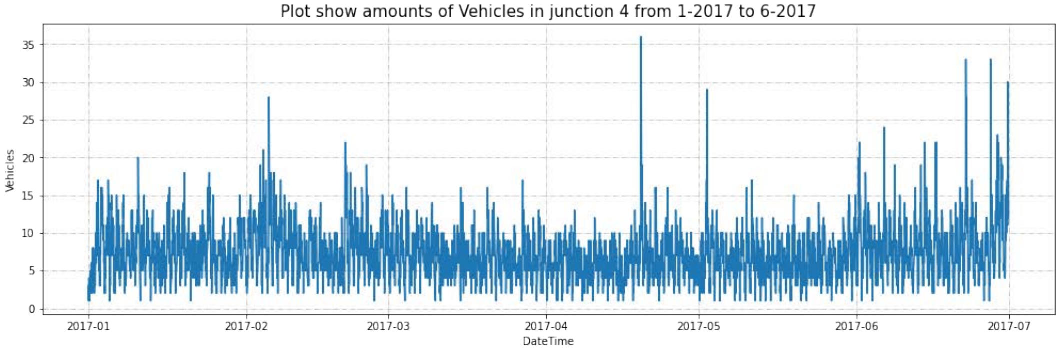
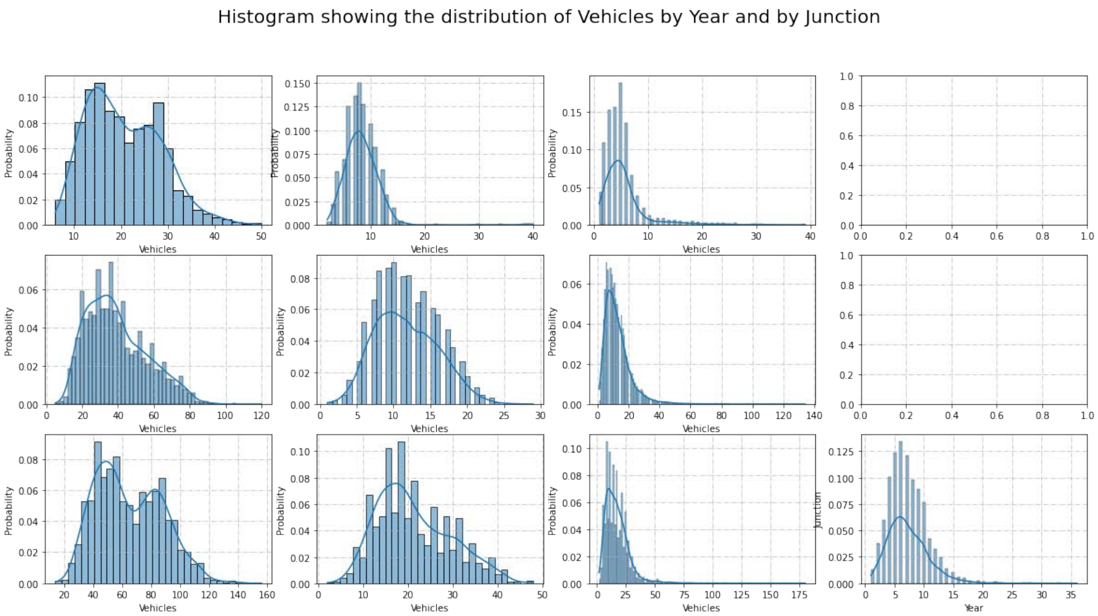


Figure 3.5: The amounts of vehicles in junction 4

And, there is a grid of histograms, where each histogram represents the distribution of vehicle counts over different years and junctions.



Histogram showing the distribution of Vehicles by Year and by Junction

Figure 3.6: The distribution of Vehicles by Year and by Junction

In understanding traffic dynamics and seasonality these visualizations highlight temporal patterns and fluctuations in traffic volume.

Normalization and Outlier Detection:
For better comparison and analysis across junctions and periods Standardization of data using Z-score normalization is allowed.

DateTime	**Junction**	**Vehicles**	**Year**	**Month**	**Day**	**Hour**
2015-11-01 00:00:00	1	-0.375489	2015	11	1	0
2015-11-01 01:00:00	1	-0.471875	2015	11	1	1
2015-11-01 02:00:00	1	-0.616454	2015	11	1	2
2015-11-01 03:00:00	1	-0.761034	2015	11	1	3
2015-11-01 04:00:00	1	-0.664648	2015	11	1	4

Table 3.3: The first five rows of the modified DataFrame made by the standardization process

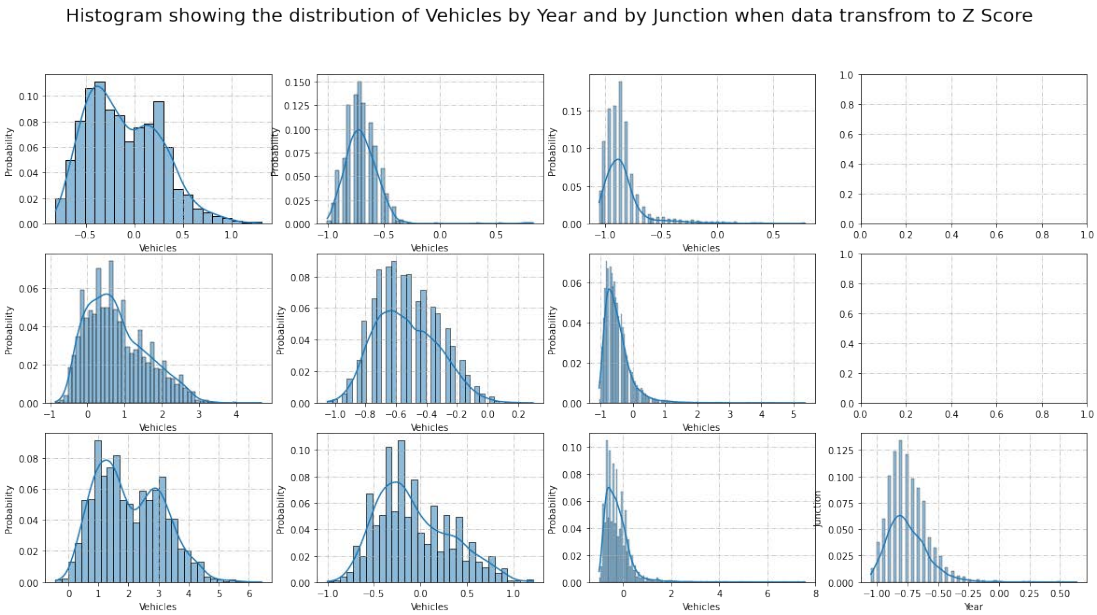


Figure 3.7: The distribution of Vehicles by Year and by Junction from transformed Dataframe

Correlation Analysis:

Different attributes, revealing relationships such as the correlation between year and month, and identifying potential multicollinearity issues are visualized the correlation by Heatmaps.

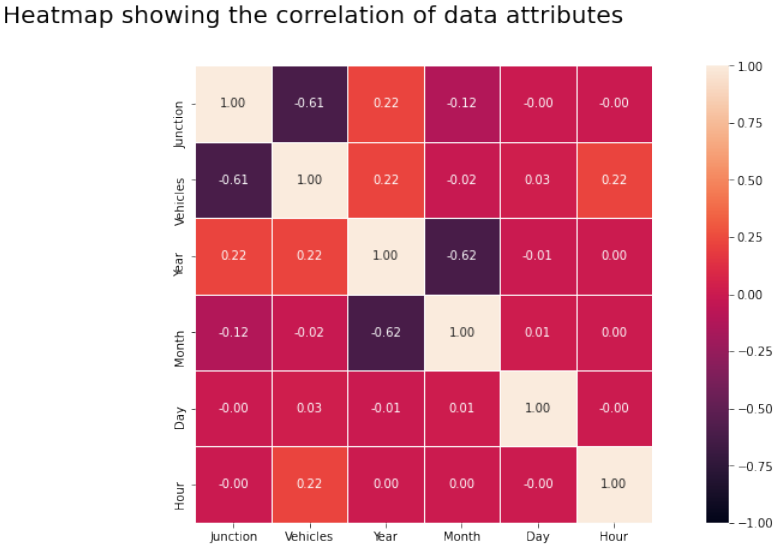


Figure 3.8: The correlation of data attributes

In feature selection and model building, ensuring that redundant or highly correlated features are not included while understanding correlation.

Further Data Analysis by Dropping the Specified Columns:

For further analysis and visualization of vehicle count data across different junctions "get_list_data" function is created. It creates a list of DataFrames, each representing data for a different junction, filtering rows based on junction numbers and removing the "Junction" column, then returns the list of DataFrames also drops specified columns from the data frame. For focused analysis and visualization tailored to each junction's traffic patterns and trends, it helps to enable easy access and manipulation of data for individual junctions.

<table>
<tr><td colspan="6">Empty DataFrame
Columns: [Vehicles, Year, Month, Day, Hour]
Index:[]</td></tr>
<tr><td>DateTime</td><td>Vehicles</td><td>Year</td><td>Month</td><td>Day</td><td>Hour</td></tr>
<tr><td>2015-11-01</td><td>15</td><td>2015</td><td>11</td><td>1</td><td>0</td></tr>
<tr><td>2015-11-01</td><td>6</td><td>2015</td><td>11</td><td>1</td><td>0</td></tr>
<tr><td>2015-11-01</td><td>9</td><td>2015</td><td>11</td><td>1</td><td>0</td></tr>
<tr><td>2017-01-01</td><td>3</td><td>2017</td><td>1</td><td>1</td><td>0</td></tr>
</table>

Table 3.4: The first row of each DataFrame in the list

From the list of DataFrame, some line plots are given for understanding the variations in traffic volume over the months and even finer time intervals like days.

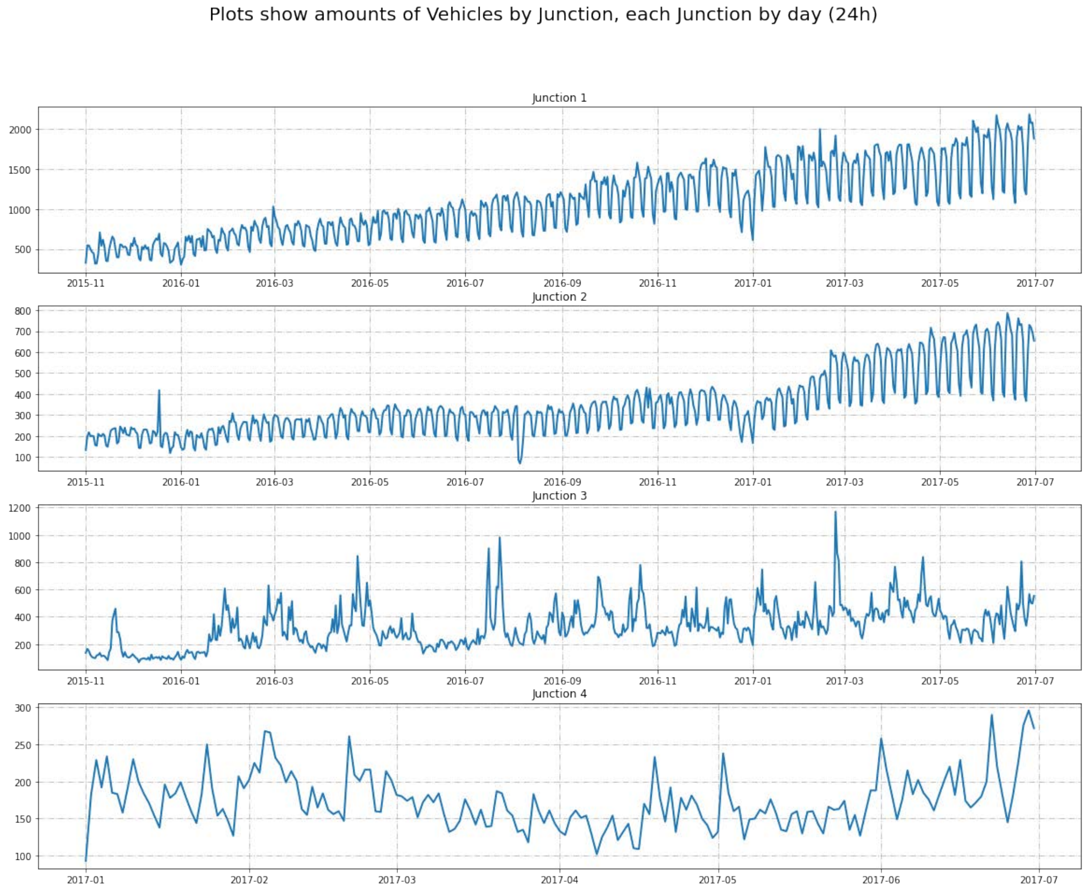


Figure 3.9: The amounts of Vehicles by Junction, each Junction by day (24h)

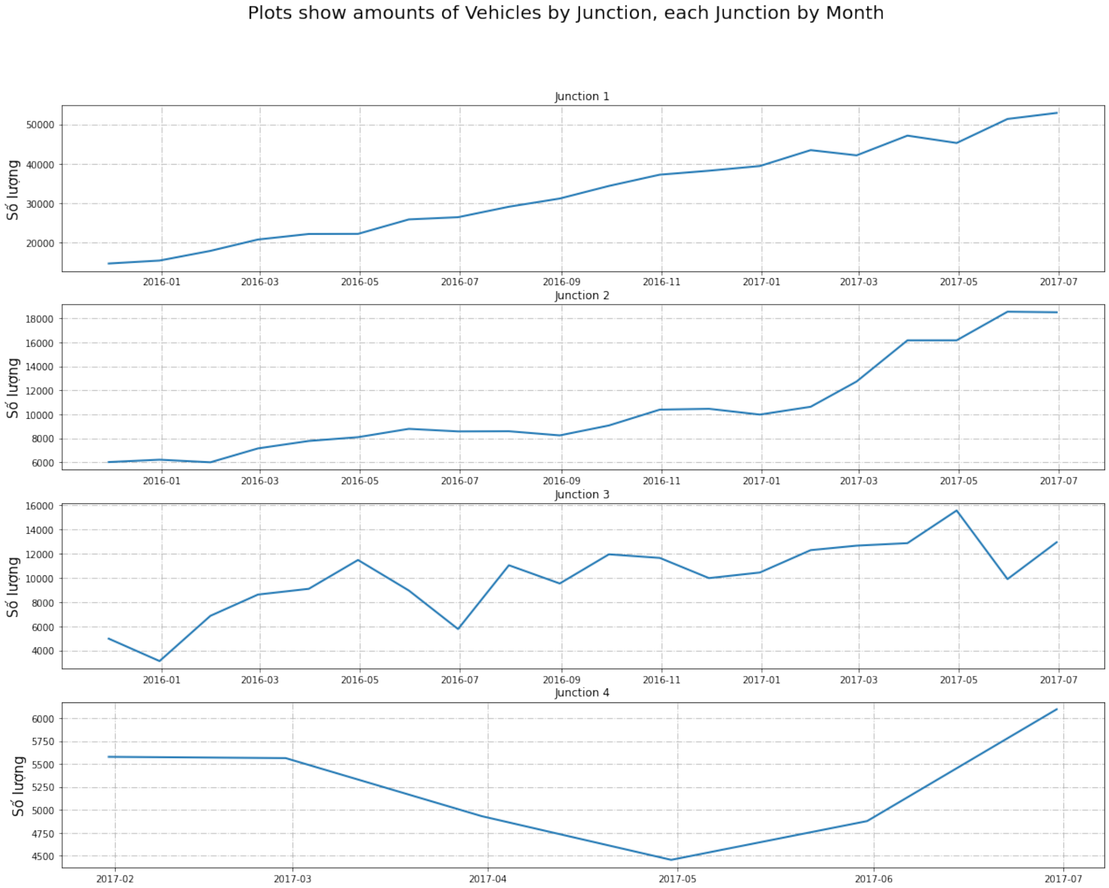


Figure 3.10: The amounts of Vehicles by Junction, each Junction by Month

Time Series Analysis:

The temporal dependencies and lag effects present in the data are provided by autocorrelation and partial autocorrelation plots.

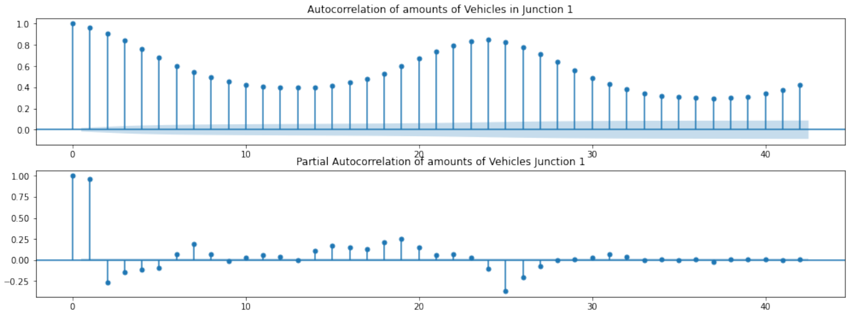


Figure 3.11: Autocorrelation and Partial Autocorrelation of amounts of Vehicles Junction 1

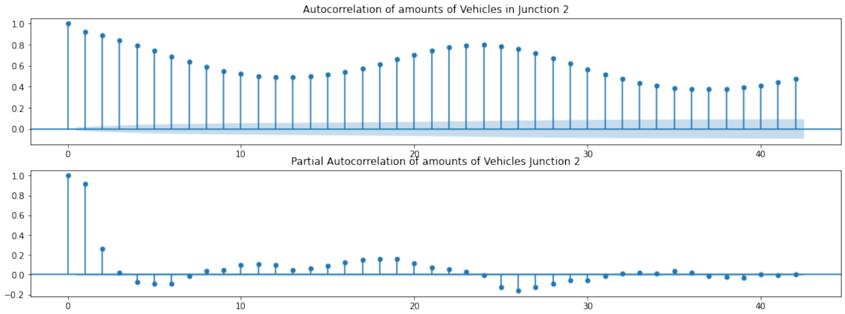


Figure 3.12: Autocorrelation and Partial Autocorrelation of amounts of Vehicles Junction 2

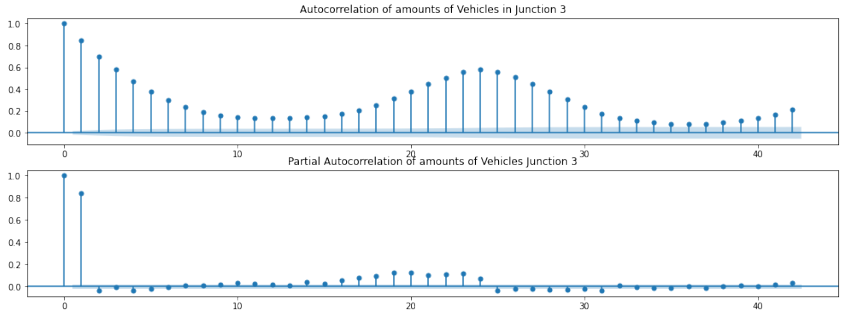


Figure 3.13: Autocorrelation and Partial Autocorrelation of amounts of Vehicles Junction 3

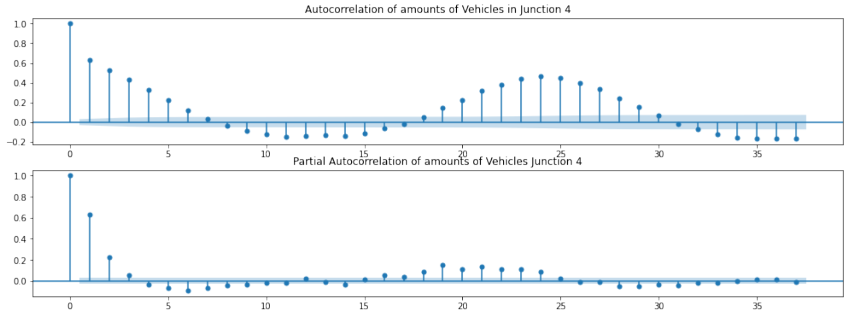


Figure 3.14: Autocorrelation and Partial Autocorrelation of amounts of Vehicles Junction 4

In time series forecasting, lag data—previous observations incorporated as features—can help identify patterns across time and enhance prediction accuracy.

The importance of understanding temporal patterns, correlations, and outliers in traffic data is underscored by the analysis. Even with the difficulties caused by temporal dynamics and outliers, machine learning approaches are useful for precisely predicting traffic volume. In conclusion, the dataset offers a wealth of data for comprehending the dynamics and patterns of traffic at various intersections. Stakeholders can obtain practical insights to improve transportation infrastructure and traffic management through thorough study and modeling.

3.3 Data Preprocessing

For training machine learning models, the data preprocessing step is super important. By bundling different preprocessing tasks specifically designed for the traffic dataset, the model class comes in handy.

Initialization and Dataset Overview:
You need to give it important information like the dataset, what you're trying to predict (target variable), how much of the data you want to use for testing, and what type of machine learning model you're using when you create an instance of the Model class. In the CSV file, the dataset stored is all about traffic. When each piece of data was recorded it included details about vehicles and had a DateTime index to show. The target variable is the number of vehicles, which tells us how much traffic there is at a given time. This is what we're trying to predict with our machine-learning model.

Extracting Temporal Features:
From the DateTime index, the initial preprocessing step focuses on extracting temporal features. We isolate components like Year, Month, Day, and Hour to help the model better understand and capture temporal patterns in the data. How traffic volumes change over various time scales because of these features is vital for recognizing. Making it both efficient and straightforward, we leverage Pandas' datetime functionalities ans streamline the feature extraction process.

Dropping Unnecessary Columns:
To simplify the dataset and eliminate any irrelevant Information we choose to drop the 'ID' column. 'ID' column doesn't add significant value to the predictive capabilities of the traffic dataset. To pass this problem pandas provide a method called drop.

Standardization:
A lambda function called standardization is created to standardize the 'Vehicles' column. We use the scikit-learn library's StandardScaler in this lambda function. Standardization is a process used to transform the data to have a mean of 0 and a standard deviation of 1. This ensures consistency and comparability.

DataFrame Copy and Transformation:
We refer to as 'z_df' where we make a duplicate of the original DataFrame df. After that by applying the standardization function that was previously defined we standardize the 'Vehicles' column in 'z_df'. This guarantees that the numbers in the 'Vehicles' column are scaled according to the standardization procedure, which uses a mean of 0 and a standard deviation of 1.

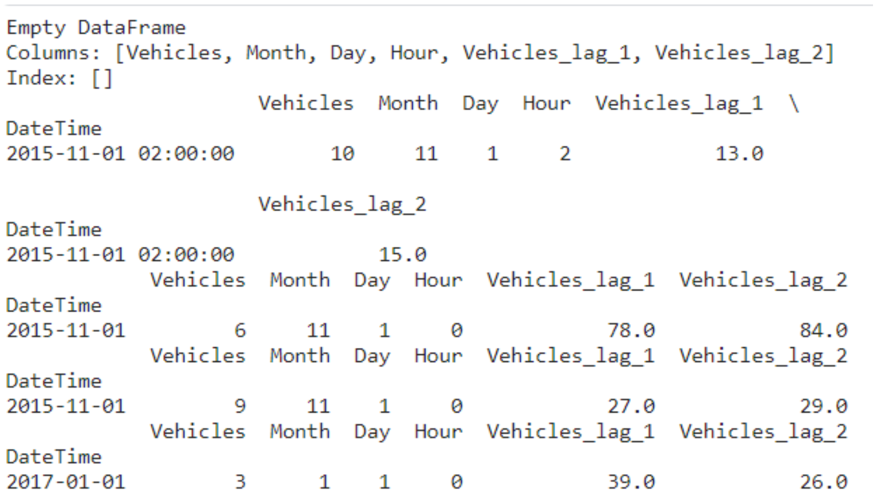
Data Segmentation by Junction:
A list of columns to drop (drop) when the function get_list_data accepts a DataFrame (dataf). It creates a list of DataFrames containing data specific to each junction after iterating through each junction (indexed from 0 to 4). During this process, it removes the 'Junction' column from each DataFrame. As a result, this segment analyzes data based on different junctions independently.

Lag Feature Creation:

The DataFrame lag_df is created as a copy of the original DataFrame df. Lag features, denoted as ‘Vehicles_lag_1’ and ‘Vehicles_lag_2’, are generated by shifting the ‘Vehicles’ column by 1 and 2 time steps, respectively. This allows capturing temporal patterns and dependencies in the data.

DateTime	Junction	Vehicles	Year	Month	Day	Hour	Vehicles_lag_1	Vehicles_lag_2
2015-11-01 02:00:00	1	10	2015	11	1	2	13.0	15.0
2015-11-01 03:00:00	1	7	2015	11	1	3	10.0	13.0
2015-11-01 04:00:00	1	9	2015	11	1	4	7.0	10.0
2015-11-01 05:00:00	1	6	2015	11	1	5	9.0	7.0
2015-11-01 06:00:00	1	9	2015	11	1	6	6.0	9.0

Table 3.5: The description of first five rows of the Lag Data



```
Empty DataFrame
Columns: [Vehicles, Month, Day, Hour, Vehicles_lag_1, Vehicles_lag_2]
Index: []
                     Vehicles  Month  Day  Hour  Vehicles_lag_1  \
DateTime
2015-11-01 02:00:00        10     11    1     2            13.0

                     Vehicles_lag_2
DateTime
2015-11-01 02:00:00            15.0
            Vehicles  Month  Day  Hour  Vehicles_lag_1  Vehicles_lag_2
DateTime
2015-11-01         6     11    1     0            78.0            84.0
            Vehicles  Month  Day  Hour  Vehicles_lag_1  Vehicles_lag_2
DateTime
2015-11-01         9     11    1     0            27.0            29.0
            Vehicles  Month  Day  Hour  Vehicles_lag_1  Vehicles_lag_2
DateTime
2017-01-01         3      1    1     0            39.0            26.0
```

Figure 3.15: The description of first row of each DataFrame in the Lag Data list

Train-Test Split:

The dataset is split into training and testing sets using the ‘train_test_split’ function from Scikit-Learn. The ‘test_size’ parameter, which was provided during class initialization, specifies the proportion of the dataset allocated for testing. This step ensures that the performance of the model can be properly evaluated on unseen data, helping to gauge its generalization capability.

3.4 Implementation

In this section, we look at the Model class's main functions, such as training, evaluating, analysing the importance of features, and making predictions. The layout of the code focuses on flexibility, which makes it easy to combine different machine learning models and smart analyses.

Model Training and Fitting:
The model training method is the most important part of the execution. The fit method uses the training data to teach the given machine learning model what to do. In this case, the Random Forest Regressor is stored in the class property ml_model. The model can now make predictions on the test set after it has been taught.

Evaluation Metrics Calculation:
The Root Mean Squared Error (RMSE) and the R^2 score are two evaluation metrics that are used to measure how well the model works. The RMSE, which is found by using Scikit-Learn's mean_squared_error with squared=False, shows how well the model predicted in the original units (vehicles). The R^2 score, which is found using Scikit-Learn's r2_score, measures how much of the variation can be explained by the model.

Feature Importance Analysis:
It is very important to understand which features help the model make predictions so that the forecasts can be understood. The feature_importances method makes it easier to see which features have the most impact on the model's output. This study is especially helpful for people who want to understand what makes traffic estimates possible.

Forecasting Future Traffic:
The do things method manages the whole process, from preparing the data to training the model and evaluating it. The method also makes it possible for the model to predict how much traffic will be in the next four months. The model gives us a look at how traffic might change in the future using old data and patterns from the past.

Visualization for Exploration and Communication:
In order to help with both studying and talking about traffic trends the system has many visualizations. To get a better sense of the information as a whole we use histograms, time series plots, and visualizations of feature value. To better understand how traffic moves and changes over time and space the researchers and other interested parties can use these visualizations

To sum up, without any problems the Model class application combines data preparation, model training, evaluation, and predictions. It's easy to try out different machine learning models with the flexible design, and the model can be understood with the help of helpful visualisations. This all-encompassing method helps us learn more about how vehicle travel works. The adaptable design makes it simple to test out various machine learning models, and the model's useful visuals make it easy to understand. This comprehensive approach contributes to our understanding of the mechanics of vehicle travel.The adaptable design makes it simple to test out various machine learning models, and the model's useful visuals make it easy to understand. This comprehensive approach contributes to our understanding of the mechanics of vehicle travel.

Chapter 4

Result Analysis

In this section, we will get to know how the model works efficiently and how it associates in the real-world. While investigating the model's performance measures, we can get to know how easy it works and will reach to the conclusion whether it fulfills our objectives or not.

Performance Metrics and Model Evaluation:
There are two key metrics in this model which are: Root Mean Squared Error (RMSE) and R^2 Score. Much pieces of knowledge into the model's predictive accuracy and its capability to prevail the variability in the target variable are provided by these metrics.

The average size of the variations between the predicted and actual values is determined using the cal_rmse_method, which is derived from the RMSE. A lower RMSE suggests higher predictive accuracy, as it means the model's predictions are closer to the actual values. Where else, measures the proportion of the variability in the dependent variable that can be explained by the independent variables, the R^2 Score, derived from the cal_r2_score method. It can better predict the outcome variable based on the input variables when a higher R^2 Score indicates a stronger fit of the model to the data.

	name	**r^2**	**rmse**
0	average R^2 and sum RMSE	0.944508	5.477258
1	average R^2 and sum RMSE	0.861060	2.825703
2	average R^2 and sum RMSE	0.747403	5.176516
3	average R^2 and sum RMSE	0.477476	2.393279
4	average R^2 and sum RMSE	0.757612	15.872757

Table 4.1: Results after training models for 4 junction with normal data

	name	r^2	rmse
0	average R^2 and sum RMSE	0.940894	0.268887
1	average R^2 and sum RMSE	0.866175	0.132760
2	average R^2 and sum RMSE	0.712885	0.281497
3	average R^2 and sum RMSE	0.462286	0.127454
4	average R^2 and sum RMSE	0.745560	0.810597

Table 4.2: Results after training models for 4 junction with Z Score Normalization

	name	r^2	rmse
0	average R^2 and sum RMSE	0.965358	4.329008
1	average R^2 and sum RMSE	0.887861	2.488898
2	average R^2 and sum RMSE	0.730946	5.408623
3	average R^2 and sum RMSE	0.496654	2.534998
4	average R^2 and sum RMSE	0.770205	14.761527

Table 4.3: Results after training models for 4 junction with Lag data

However, we constructed different models to compare with our chosen model. So we used alternative models such as SVR (Support Vector Regression), Linear Regression, KNeighborsRegressor, and Ridge Regression. The above models were constructed using lag data, which is significant in time series research because it helps models to capture temporal relationships, spot trends, and make accurate predictions about future values based on prior knowledge.

While applying the other models, we attempted to compare the outcomes (r2 and rmse values) to determine which model performed better. We compared our chosen model to other models independently to have a better understanding.

Table 4.4 compares the Random Forest Regressor and Support Vector Regression (SVR). Across all junctions, Random Forest Regressor had higher R^2 scores, demonstrating that the data fits better to the regression model than SVR. In addition, when compared to SVR, the Random Forest Regressor produces lower RMSE values, indicating smaller prediction errors. This shows that the Random Forest Regressor catches underlying patterns in data more effectively than SVR, giving it a better alternative for forecasting traffic tasks.

	Random Forest Regressor		SVR	
	r^2	rmse	r^2	rmse
0	0.965358	4.329008	0.935476	5.819115
1	0.887861	2.488898	0.869258	2.652224
2	0.730946	5.408623	0.655793	6.288105
3	0.496654	2.534998	0.501245	2.478529
4	0.770205	14.761527	0.740443	17.237973

Table 4.4: Comparison of Model Performance between Random Forest Regressor and SVR for 4 Junctions with Lag Data

In Table 4.5, Random Forest Regressor is compared to Linear Regression. Once again, the Random Forest Regressor outperforms the other models, as seen by higher R^2 scores and lower RMSE values at every junction. Which simplifies that compared to Linear Regression, the Random Forest Regressor acquires nonlinear relationships and complexities in the data more specifically means more precise predictions.

	Random Forest Regressor		Linear Regression	
	r^2	**rmse**	**r^2**	**rmse**
0	0.965358	4.329008	0.932577	5.941119
1	0.887861	2.488898	0.844478	2.951809
2	0.730946	5.408623	0.724400	5.645706
3	0.496654	2.534998	0.457772	2.635472
4	0.770205	14.761527	0.739807	17.174105

Table 4.5: Comparison of Model Performance between Random Forest Regressor and Linear Regression for 4 Junctions with Lag Data

In Table 4.6, the comparison between Random Forest Regressor and KNeighborsRegressor shows us that Random Forest Regressor constantly surpasses KNeighborsRegressor if we notice the R^2 and the RMSE values. This points out that Random Forest Regressor is better than KNeighborsRegressor in terms of capturing the elaborate connections among the data which means more precise predictions.

	Random Forest Regressor		KNeighborsRegressor	
	r^2	**rmse**	**r^2**	**rmse**
0	0.965358	4.329008	0.953886	4.946739
1	0.887861	2.488898	0.871243	2.703941
2	0.730946	5.408623	0.725900	5.307777
3	0.496654	2.534998	0.434728	2.790038
4	0.770205	14.761527	0.746439	15.748494

Table 4.6: Comparison of Model Performance between Random Forest Regressor and KNeighborsRegressor for 4 Junctions with Lag Data

Looking into the comparison between Random Forest Regressor and Ridge Regression in terms of R^2 scores and RMSE values here in Table 4.7 indicates that Random Forest Regressor outperforms Ridge Regression clearly which brings out that compared to Ridge Regression, Random Forest Regressor seizes the complexities of the data and produces predictions more effectively.

	Random Forest Regressor		Ridge Regression	
	r^2	**rmse**	**r^2**	**rmse**
0	0.965358	4.329008	0.934479	5.910349
1	0.887861	2.488898	0.862593	2.778234
2	0.730946	5.408623	0.696627	5.806019
3	0.496654	2.534998	0.418662	2.731525
4	0.770205	14.761527	0.728090	17.226126

Table 4.7: Comparison of Model Performance between Random Forest Regressor and Ridge Regression for 4 Junctions with Lag Data

In summary, at every junction, the Random Forest Regressor outperforms the other regression models we have tested continuously and this means that its implementation could be convenient in terms of lag data-based traffic prediction tasks. Its capacity to capture nonlinear correlations and manage complicated information makes it a strong candidate for such predictive modeling projects.

Feature Importance and Model Interpretability:

It is essential to comprehend the significance of features in order to interpret the model. The feature_importances approach aids in identifying the elements influencing the model's predictions by visualizing the most significant characteristics. Stakeholders can quickly assess the relative importance of each element and determine which factors influence traffic volume by using a bar plot.

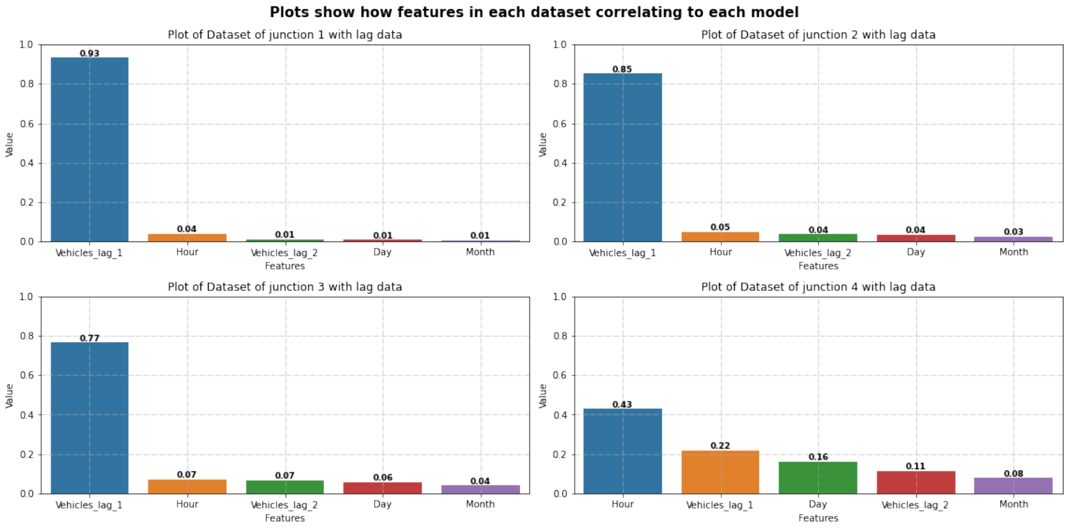


Figure 4.1: Features in each dataset correlating to each model

Here, in junction 1, recent traffic volume strongly affects current traffic conditions indicating the high correlation coefficient (0.93) for Vehicles_lag_1. However, some features like Hour, Day, and Month reveal lower correlations (0.04, 0.01, and 0.01, respectively). There are some impacts of the time of day or particular calendar on traffic volume if we compare recent historical data.

Again, in junction 2, Vehicles_lag_1 sets a high correlation coefficient (0.85) which suggests its major effect on predicting traffic trends in junction 2. Moreover, Hour and Vehicles_lag_2 also reveal noticeable correlations (0.05 and 0.04, respectively). Both current and recent historical traffic data are necessary in order to make accurate predictions. Day and Month features have partially lower correlations (0.04 and 0.03, respectively) which implies that they have little influence compared to time-related and historical traffic data.

In junction 3, Vehicles_lag_1 again is a significant predictor with a high correlation coefficient (0.75). Both Hour and Vehicles_lag_2 exhibit similar correlations (0.07 each) which defines their significance in predicting traffic patterns. Day and Month features have comparatively lower correlations (0.06 and 0.05, respectively). Temporal factors roughly influence traffic predictions In this junction.

Nevertheless, the most important attribute is hour, which has a strong correlation coefficient (0.42) in junction 4. This denotes that the time of day firmly regulates traffic volume and it is probably because of rush hours or specific commuting patterns. The moderate association (0.24) between Vehicles_lag_1 and their pursuit indicates that it may have implications for recording real-time traffic trends. Day and Vehicles lag 2 have moderate correlations (0.15 and 0.12, respectively), but Month has a lesser connection (0.07). This implies that the daily fluctuations and recent historical data have a significant role in forecasting Junction 4 traffic numbers.

The following important inferences about the anticipated outcome for traffic volume across junctions can be made based on the correlation analysis of features:

The most crucial element in predicting the current traffic condition is the traffic from the previous hour (Vehicles_lag_1). This highlights the importance of short-term historical data in forecasting by showing how recent traffic patterns have a significant influence on future traffic figures.

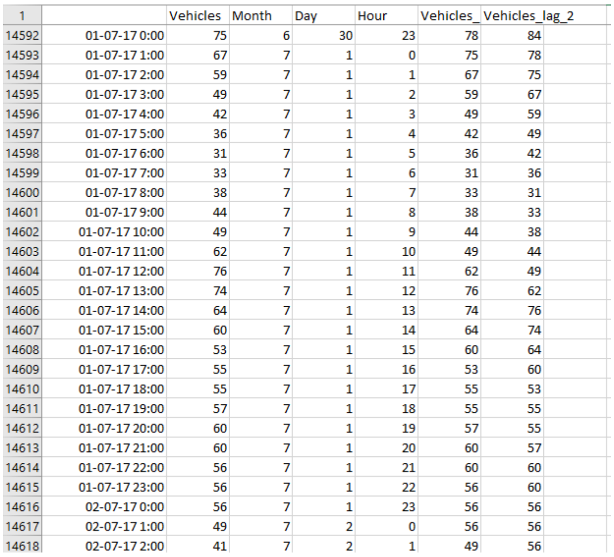
Vehicles_lag_1, which consistently shows a high positive connection with traffic forecast outcomes, is more significant at different junctions than other factors, such as the hour, the day, the month, and Vehicles_lag_2 (traffic from the previous hour). The complexity of traffic patterns can be emphasized by this fluctuation, which varies based on the location of each intersection, the time of day, and the seasons.

Significant predictive power is seen when correlation values are near to 1, which suggests a strong positive link between the feature and the anticipated outcome. Conversely, values nearer zero signify weaker connections and, hence, less importance for traffic forecasting.

Creating predictive algorithms that perform well requires an understanding of the critical components that go into accurate traffic forecasts at each intersection. By employing parameters such as Vehicles_lag_1 with large correlation values, models can more accurately capture the underlying patterns and dynamics of traffic behavior, leading to improved forecast accuracy.

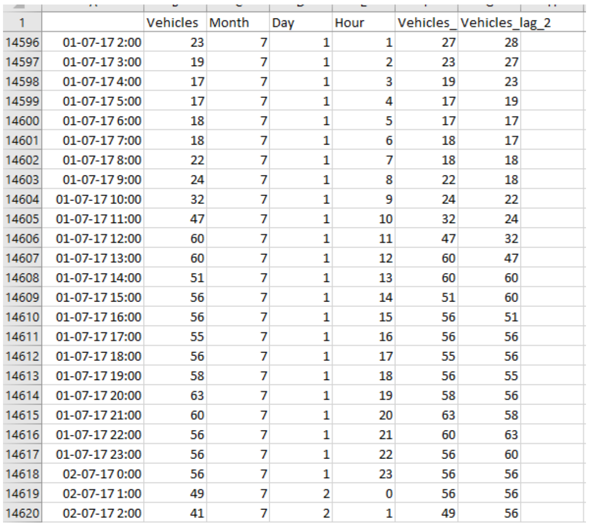
To sum up, the correlation analysis offers useful insights into the attributes that are most important for predicting traffic volumes at different crossings. By identifying and utilizing these critical signs, developers can modify their approaches to fit the distinct features of each junction. This eventually improves the efficiency and dependability of traffic prediction algorithms.

Future Traffic Forecasting:
The implemented model does more than just research in the past; it predicts future traffic volumes. Using previous patterns and lagged data, it employs the “do things” approach to anticipate traffic trends for the ensuing four months. Next, a CSV file specific to each intersection contains the expected number of vehicles over the next four months at that intersection. For the purpose of managing infrastructure, urban planning, and forecasting future traffic congestion, this forward-looking perspective is quite beneficial.



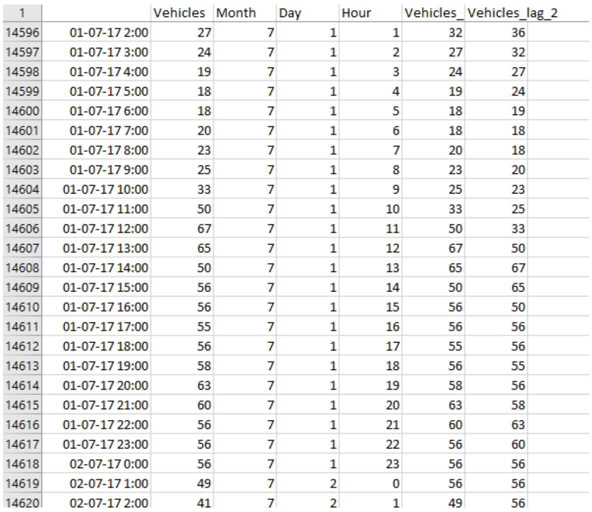
1		Vehicles	Month	Day	Hour	Vehicles_	Vehicles_lag_2
14592	01-07-17 0:00	75	6	30	23	78	84
14593	01-07-17 1:00	67	7	1	0	75	78
14594	01-07-17 2:00	59	7	1	1	67	75
14595	01-07-17 3:00	49	7	1	2	59	67
14596	01-07-17 4:00	42	7	1	3	49	59
14597	01-07-17 5:00	36	7	1	4	42	49
14598	01-07-17 6:00	31	7	1	5	36	42
14599	01-07-17 7:00	33	7	1	6	31	36
14600	01-07-17 8:00	38	7	1	7	33	31
14601	01-07-17 9:00	44	7	1	8	38	33
14602	01-07-17 10:00	49	7	1	9	44	38
14603	01-07-17 11:00	62	7	1	10	49	44
14604	01-07-17 12:00	76	7	1	11	62	49
14605	01-07-17 13:00	74	7	1	12	76	62
14606	01-07-17 14:00	64	7	1	13	74	76
14607	01-07-17 15:00	60	7	1	14	64	74
14608	01-07-17 16:00	53	7	1	15	60	64
14609	01-07-17 17:00	55	7	1	16	53	60
14610	01-07-17 18:00	55	7	1	17	55	53
14611	01-07-17 19:00	57	7	1	18	55	55
14612	01-07-17 20:00	60	7	1	19	57	55
14613	01-07-17 21:00	60	7	1	20	60	57
14614	01-07-17 22:00	56	7	1	21	60	60
14615	01-07-17 23:00	56	7	1	22	56	60
14616	02-07-17 0:00	56	7	1	23	56	56
14617	02-07-17 1:00	49	7	2	0	56	56
14618	02-07-17 2:00	41	7	2	1	49	56

Figure 4.2: First few rows of Predicted vehicle counts for next 4 months in Junction 1



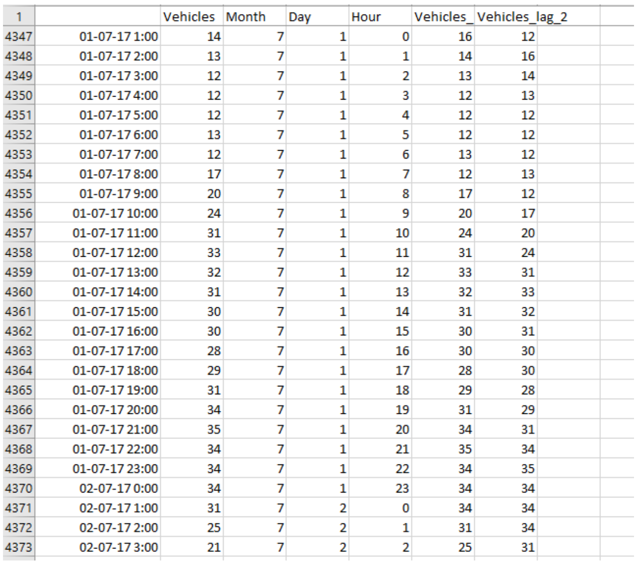
1		Vehicles	Month	Day	Hour	Vehicles_	Vehicles_lag_2
14596	01-07-17 2:00	23	7	1	1	27	28
14597	01-07-17 3:00	19	7	1	2	23	27
14598	01-07-17 4:00	17	7	1	3	19	23
14599	01-07-17 5:00	17	7	1	4	17	19
14600	01-07-17 6:00	18	7	1	5	17	17
14601	01-07-17 7:00	18	7	1	6	18	17
14602	01-07-17 8:00	22	7	1	7	18	18
14603	01-07-17 9:00	24	7	1	8	22	18
14604	01-07-17 10:00	32	7	1	9	24	22
14605	01-07-17 11:00	47	7	1	10	32	24
14606	01-07-17 12:00	60	7	1	11	47	32
14607	01-07-17 13:00	60	7	1	12	60	47
14608	01-07-17 14:00	51	7	1	13	60	60
14609	01-07-17 15:00	56	7	1	14	51	60
14610	01-07-17 16:00	56	7	1	15	56	51
14611	01-07-17 17:00	55	7	1	16	56	56
14612	01-07-17 18:00	56	7	1	17	55	56
14613	01-07-17 19:00	58	7	1	18	56	55
14614	01-07-17 20:00	63	7	1	19	58	56
14615	01-07-17 21:00	60	7	1	20	63	58
14616	01-07-17 22:00	56	7	1	21	60	63
14617	01-07-17 23:00	56	7	1	22	56	60
14618	02-07-17 0:00	56	7	1	23	56	56
14619	02-07-17 1:00	49	7	2	0	56	56
14620	02-07-17 2:00	41	7	2	1	49	56

Figure 4.3: First few rows of Predicted vehicle counts for next 4 months in Junction 2



1		Vehicles	Month	Day	Hour	Vehicles_	Vehicles_lag_2
14596	01-07-17 2:00	27	7	1	1	32	36
14597	01-07-17 3:00	24	7	1	2	27	32
14598	01-07-17 4:00	19	7	1	3	24	27
14599	01-07-17 5:00	18	7	1	4	19	24
14600	01-07-17 6:00	18	7	1	5	18	19
14601	01-07-17 7:00	20	7	1	6	18	18
14602	01-07-17 8:00	23	7	1	7	20	18
14603	01-07-17 9:00	25	7	1	8	23	20
14604	01-07-17 10:00	33	7	1	9	25	23
14605	01-07-17 11:00	50	7	1	10	33	25
14606	01-07-17 12:00	67	7	1	11	50	33
14607	01-07-17 13:00	65	7	1	12	67	50
14608	01-07-17 14:00	50	7	1	13	65	67
14609	01-07-17 15:00	56	7	1	14	50	65
14610	01-07-17 16:00	56	7	1	15	56	50
14611	01-07-17 17:00	55	7	1	16	56	56
14612	01-07-17 18:00	56	7	1	17	55	56
14613	01-07-17 19:00	58	7	1	18	56	55
14614	01-07-17 20:00	63	7	1	19	58	56
14615	01-07-17 21:00	60	7	1	20	63	58
14616	01-07-17 22:00	56	7	1	21	60	63
14617	01-07-17 23:00	56	7	1	22	56	60
14618	02-07-17 0:00	56	7	1	23	56	56
14619	02-07-17 1:00	49	7	2	0	56	56
14620	02-07-17 2:00	41	7	2	1	49	56

Figure 4.4: First few rows of Predicted vehicle counts for next 4 months in Junction 3



1		Vehicles	Month	Day	Hour	Vehicles_	Vehicles_lag_2
4347	01-07-17 1:00	14	7	1	0	16	12
4348	01-07-17 2:00	13	7	1	1	14	16
4349	01-07-17 3:00	12	7	1	2	13	14
4350	01-07-17 4:00	12	7	1	3	12	13
4351	01-07-17 5:00	12	7	1	4	12	12
4352	01-07-17 6:00	13	7	1	5	12	12
4353	01-07-17 7:00	12	7	1	6	13	12
4354	01-07-17 8:00	17	7	1	7	12	13
4355	01-07-17 9:00	20	7	1	8	17	12
4356	01-07-17 10:00	24	7	1	9	20	17
4357	01-07-17 11:00	31	7	1	10	24	20
4358	01-07-17 12:00	33	7	1	11	31	24
4359	01-07-17 13:00	32	7	1	12	33	31
4360	01-07-17 14:00	31	7	1	13	32	33
4361	01-07-17 15:00	30	7	1	14	31	32
4362	01-07-17 16:00	30	7	1	15	30	31
4363	01-07-17 17:00	28	7	1	16	30	30
4364	01-07-17 18:00	29	7	1	17	28	30
4365	01-07-17 19:00	31	7	1	18	29	28
4366	01-07-17 20:00	34	7	1	19	31	29
4367	01-07-17 21:00	35	7	1	20	34	31
4368	01-07-17 22:00	34	7	1	21	35	34
4369	01-07-17 23:00	34	7	1	22	34	35
4370	02-07-17 0:00	34	7	1	23	34	34
4371	02-07-17 1:00	31	7	2	0	34	34
4372	02-07-17 2:00	25	7	2	1	31	34
4373	02-07-17 3:00	21	7	2	2	25	31

Figure 4.5: First few rows of Predicted vehicle counts for next 4 months in Junction 4

The accuracy with which these forecasts are produced demonstrates the model's capacity to recognise underlying patterns and trends. Stakeholders can utilize these projections to maximize traffic flow, plan ahead, and allocate resources efficiently in order to tackle any issues.

Temporal Pattern Analysis:
Complex time patterns can be found in traffic statistics. The Year, Month, Day, and Hour attributes are extracted by the model to represent this intricacy. This enables a more thorough analysis of these patterns. Use line plots and histograms made using the "make_time_series_plot" tool to provide users with a visual depiction of the swings in traffic volume across various time scales.

The time series plots give stakeholders a live picture of the traffic trends at each intersection, making it possible to spot recurrent patterns and abnormalities. Histograms aid in further illuminating the traffic volume distribution by showing which crossings have normal or skew distributions.

Correlation Heatmap:

The correlation heatmap—which was made possible by the Pandas and Seaborn libraries—improves the study. It accomplishes this by emphasizing the connections between the various dataset variables. Through the presentation of correlations, interested parties can determine possible relationships between various variables. The heatmap highlights characteristics that exhibit strong positive or negative correlations, which helps to clarify the interactions between different components.

Chapter 5

Conclusion

In summary, an effective and educational tool for analyzing, forecasting, and improving traffic patterns is offered by the developed model and analysis framework. Using Python modules like Seaborn, Pandas, and Scikit-Learn, this modular design provides flexibility, scalability, and ease of interpretation. Temporal insights and forecasting capabilities, accessibility and stakeholder communication, visualization for exploratory data analysis (EDA), and model diversity and adaptation are the main areas of concern for the findings.

The underlying machine learning model, called the "Random Forest Regressor," demonstrates its adaptability. It is flexible enough to accommodate different traffic projection scenarios and has a large dataset handling capacity. Customers may select the best approach for their particular needs and replace models more readily with the aid of this framework.

The ability of the model to gather and process temporal data allows for a deeper understanding of traffic patterns. The model collects and uses temporal dynamics, ranging from daily changes to monthly trends, to produce projections that are more accurate. By including a forecasting mechanism for the next four months, the model's utility is increased beyond that of earlier research, making it a valuable tool for proactive decision-making.

Our methodology places a strong emphasis on model interpretability. Interpretability is crucial for findings to be communicated to stakeholders who may not be experts in machine learning. These findings can have an impact on how policies, attitudes, and resources are distributed.

Several visualization techniques are used to improve the code's exploratory data analysis (EDA) capabilities. These visualizations, which can include time series plots that display temporal patterns or histograms that display data distributions, enable users to explore the dataset from a number of perspectives. The correlation heatmap adds a sophisticated layer to the EDA process by highlighting relationships between data that were previously missed.

Good data science necessitates continuous development, even in cases where the deployed model performs admirably. Subsequent versions of this model could investigate ensemble techniques, hyperparameter adjustments, or even more complex time series models. Furthermore, the prediction power of the model may be improved by including external variables like events, meteorological data, or road closures.

This approach has broad implications that extend well beyond data science. The model's conclusions will be useful to legislators, traffic control organizations, and urban planners. Predicting traffic patterns, optimizing traffic flow, and strategically planning infrastructure improvements are a few of the real-world uses.

As a result, the created model is a helpful tool for the study of traffic data. The tool is an excellent resource for anyone interested in traffic management and urban planning because of its interpretability, adaptability, and projections. The model's design principles and insights enable informed decision-making in the dynamic and intricate field of urban mobility.

Bibliography

[1] M. E. Fouladvand, Z. Sadjadi, and M. R. Shaebani, "Optimized traffic flow at a single intersection: traffic responsive signalization," *Journal of physics*, vol. 37, no. 3, pp. 561–576, Jan. 2004. DOI: 10.1088/0305-4470/37/3/002. [Online]. Available: https://doi.org/10.1088/0305-4470/37/3/002.

[2] F. Teklu, A. Sumalee, and D. Watling, "A genetic algorithm approach for optimizing traffic control signals considering routing," *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, no. 1, pp. 31–43, Nov. 2006. DOI: 10.1111/j.1467-8667.2006.00468.x. [Online]. Available: https://doi.org/10.1111/j.1467-8667.2006.00468.x.

[3] K. T. K. Teo, W. Y. Kow, and Y. K. Chin, "Optimization of Traffic Flow within an Urban Traffic Light Intersection with Genetic Algorithm," *IEEE Xplore*, Sep. 2010. DOI: 10.1109/cimsim.2010.95. [Online]. Available: https://doi.org/10.1109/cimsim.2010.95.

[4] M. Treiber and A. Kesting, *Traffic flow dynamics*. Jan. 2013. DOI: 10.1007/978-3-642-32460-4. [Online]. Available: https://doi.org/10.1007/978-3-642-32460-4.

[5] E. Walraven, M. T. J. Spaan, and B. Bakker, "Traffic flow optimization: A reinforcement learning approach," *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 203–212, Jun. 2016. DOI: 10.1016/j.engappai.2016.01.001. [Online]. Available: https://doi.org/10.1016/j.engappai.2016.01.001.

[6] J. Wang, Q. Gu, J. Wu, G. Liu, and X. Zhang, "Traffic Speed Prediction and Congestion Source Exploration: A Deep Learning Method," *IEE Xplore*, Dec. 2016. DOI: 10.1109/icdm.2016.0061. [Online]. Available: https://doi.org/10.1109/icdm.2016.0061.

[7] Z. Li, M. Shahidehpour, S. Bahramirad, and A. Khodaei, "Optimizing traffic signal settings in smart cities," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2382–2393, Sep. 2017. DOI: 10.1109/tsg.2016.2526032. [Online]. Available: https://doi.org/10.1109/tsg.2016.2526032.

[8] P. Chhatpar, N. Doolani, S. Shahani, and R. L. Priya, "Machine learning solutions to vehicular traffic congestion," *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, Jan. 2018. DOI: 10.1109/icscet.2018.8537260. [Online]. Available: https://doi.org/10.1109/icscet.2018.8537260.

[9] A. Ata, M. A. Khan, S. Abbas, G. Ahmad, and A. Fatima, "MODELLING SMART ROAD TRAFFIC CONGESTION CONTROL SYSTEM USING MACHINE LEARNING TECHNIQUES," *Neural Network World*, vol. 29, no. 2, pp. 99–110, Jan. 2019. DOI: 10.14311/nnw.2019.29.008. [Online]. Available: https://doi.org/10.14311/nnw.2019.29.008.

[10] D. Nallaperuma, R. Nawaratne, T. Bandaragoda, *et al.*, "Online Incremental Machine Learning platform for Big Data-Driven smart traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4679–4690, Dec. 2019. DOI: 10.1109/tits.2019.2924883. [Online]. Available: https://doi.org/10.1109/tits.2019.2924883.

[11] R. Neuhold, F. Garolla, O. Sidla, and M. Fellendorf, "Predicting and optimizing traffic flow at toll plazas," *Transportation Research Procedia*, vol. 37, pp. 330–337, Jan. 2019. DOI: 10.1016/j.trpro.2018.12.200. [Online]. Available: https://doi.org/10.1016/j.trpro.2018.12.200.

[12] M. Akhtar and S. Moridpour, "A review of traffic congestion prediction using Artificial Intelligence," *Journal of Advanced Transportation*, vol. 2021, pp. 1–18, Jan. 2021. DOI: 10.1155/2021/8878011. [Online]. Available: https://doi.org/10.1155/2021/8878011.

[13] N. Jiang, Y. Deng, and A. Nallanathan, "Traffic Prediction and Random Access Control Optimization: Learning and Non-Learning-Based Approaches," *IEEE Communications Magazine*, vol. 59, no. 3, pp. 16–22, Mar. 2021. DOI: 10.1109/mcom.001.2000099. [Online]. Available: https://doi.org/10.1109/mcom.001.2000099.

[14] A. Navarro-Espinoza, O. R. López-Bonilla, E. E. García-Guerrero, *et al.*, "Traffic flow prediction for smart traffic lights using machine learning algorithms," *Technologies (Basel)*, vol. 10, no. 1, p. 5, Jan. 2022. DOI: 10.3390/technologies10010005. [Online]. Available: https://doi.org/10.3390/technologies10010005.

[15] R. Ritu, "Traffic Management using Machine Learning," *ResearchGate*, Jan. 2022. [Online]. Available: https://www.researchgate.net/publication/357717170_Traffic_Management_using_Machine_Learning.

[16] Y. A. Ali, M. Rafay, R. D. A. Khan, M. K. Sorn, and H. Jiang, "Traffic problems in Dhaka City: causes, effects, and solutions (Case study to develop a business model)," *OAlib*, vol. 10, no. 05, pp. 1–15, Jan. 2023. DOI: 10.4236/oalib.1109994. [Online]. Available: https://doi.org/10.4236/oalib.1109994.

[17] S. Iftikhar, M. Asim, Z. Zhang, *et al.*, "Target Detection and Recognition for traffic congestion in smart Cities using Deep Learning-Enabled UAVs: A Review and analysis," *Applied sciences*, vol. 13, no. 6, p. 3995, Mar. 2023. DOI: 10.3390/app13063995. [Online]. Available: https://doi.org/10.3390/app13063995.

[18] P. Kaushik, "Congestion Articulation Control Using Machine Learning Technique," *Amity Journal of Professional Practices*, vol. 3, no. 01, May 2023. DOI: 10.55054/ajpp.v3i01.631. [Online]. Available: https://doi.org/10.55054/ajpp.v3i01.631.

[19] A. Sroczyński and A. Czyżewski, "Road traffic can be predicted by machine learning equally effectively as by complex microscopic model," *Scientific Reports*, vol. 13, no. 1, Sep. 2023. DOI: 10.1038/s41598-023-41902-y. [Online]. Available: https://doi.org/10.1038/s41598-023-41902-y.

[20] G. Wassie, J. Ding, and Y. Wondie, "Traffic prediction in SDN for explainable QoS using deep learning approach," *Scientific Reports*, vol. 13, no. 1, Nov. 2023. doi: 10.1038/s41598-023-46471-8. [Online]. Available: https://doi.org/10.1038/s41598-023-46471-8.