Journal of Economics, Finance and Accounting Studies
ISSN: 2709-0809
DOI: 10.32996/jefas
Journal Homepage: www.al-kindipublisher.com/index.php/jefas

JEFAS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Dominance of Artificial Intelligence and Machine Learning Algorithms in Real-Time Traffic Flow prediction and Route Optimization in Autonomous Vehicles

Rejon Kumar Ray[1] ✉ Proshanta Kumar Bhowmik[2], Farhan Nasrullah[3] and Syed Ali Reza[4]
[13]MBA Business Analytics, Gannon University, Erie, PA, USA
[2]Department of Business Analytics, Trine University, Angola, IN, USA
[4]Department of Data Analytics, University of The Potomac (UOTP), Washington, USA
Corresponding Author: Rejon Kumar Ray, **E-mail**: ray015@gannon.edu

| ABSTRACT
The evolution of autonomous vehicles has elicited significant interest in understanding how real-time data may be used to provide enhanced driving experiences. This research project explored AI and Machine Learning methodologies applied for traffic forecasting and route optimization, and their implications for autonomous vehicles and urban mobility. For this project, the road traffic flow Dataset was utilized from Kaggle, containing 48,000 records of the flow of traffic, each including the following key features. In our work, we deployed some credible and well-established machine learning models: linear regression and random forest. These algorithms were separately trained by using a part of preprocessed data. The MSE for the Random Forest model was significantly lower, which means that the Random Forest Regressor had much smaller errors in estimating the volume of traffic compared to the Linear Regression model. The Random Forest Model had a high R^2 score, proving that this model explains a great deal of variance in the volume of traffic. This means that the 'model of random forest regressors' excellently fitted the data, snatching most of the important patterns and relationships between input features and target variables.

| KEYWORDS

Autonomous Vehicles; Real-time Traffic flow prediction; Route Optimization; Artificial Intelligence; Random Forest Regressors; Linear Regression.

| ARTICLE INFORMATION

ACCEPTED: 01 September 2024 **PUBLISHED:** 21 September 2024 **DOI:** 10.32996/jefas.2024.6.5.5

1. Introduction

Kumar & Saha (2020), contends that autonomous vehicles represent one of the major revolutions that contemporary transportation has undergone. An autonomous vehicle is a complex device guided by sophisticated algorithms functioning in a real-time decision-making procedure. In this emerging innovative development, AI and ML are two major components. Hernández-Mejía (2022), asserts that the possibility of predicting traffic flow and optimizing routes in real-time will contribute not only to improving the efficiency of autonomous cars but also to the overall management of traffic flow within cities. This research paper explores the AI and Machine Learning methodologies in traffic prediction and route optimization, implications for autonomous vehicles, and discusses the future trajectory of these technologies in urban mobility.

Miglani & Kumar (2019), articulates that the evolution of autonomous vehicles has spurred significant interest in understanding how real-time data may be used to provide enhanced driving experiences. The forecast for the traffic flow and optimizing the route are the two most critical elements determining the efficacy of any autonomous navigation system. As the urban population increases and traffic congestion turns into a serious issue, the need is pragmatically shifting to intelligent solutions. In this respect, AI and ML represent powerful tools because they are means by which great volumes of data can be processed to make informed

decisions in real-time (Gopi et al., 2023). This paper investigates the methodologies being used for the prediction of traffic flow and route optimization, focusing on how AI and ML are likely to shape the future concerning autonomous vehicles.

2. Literature Review of Related Works

Machine learning algorithms have consistently proven to be instrumental in predicting traffic flow. In particular, machine learning algorithms can process very large amounts of data and discern complex patterns. Classic traffic flow models, including statistical methods and time series, are never good enough to consider and handle the non-linear and stochastic nature of traffic dynamics (Jankovic, 2021). On the contrary, machine learning algorithms like ANN models, support vectors, and other deep learning models can cope with the intricacy mentioned above much more effectively.

An empirical study by Mushtaq et al. (2021) proposed a deep learning method using the Stacked Autoencoder model for traffic flow prediction where they found that it outperformed conventional methods by including the ARIMA model in capturing the intrinsic temporal dependencies and nonlinear patterns of the series in traffic flow. Empirical results indicated that deep learning models achieved around 10%-15% improvements in their models against conventions.

Comparative research by Navarro-Espinoza et al. (2017) was conducted, in which LSTM models were compared with other deep learning models, such as CNNs and a traditional feed-forward neural network for traffic speed prediction. Indeed, compared to the others, the LSTM model best captured the temporal correlations in traffic data. They used traffic data from the Caltrans PeMS and showed that LSTM models reduced the prediction errors by 20-30% in comparison to traditional methods.

Kumar et al. (2022), deployed an SVM model for short-term traffic flow prediction on urban road networks. Subsequently, the authors developed a hybrid model incorporating SVM with PSO to determine improved hyperparameters of SVM. The empirical results indicated that the model of SVM-PSO had better prediction accuracy and converged much faster compared with conventional SVM as well as other machine learning algorithms such as Random Forests (RF) and Gradient Boosting Machines (GBM).

Gopi et al. (2023), performed dynamic route planning for an urban environment by proposing an approach using Deep Reinforcement Learning. This DRL model was trained on real-world traffic conditions simulated with a deep Q-network. The experiment demonstrated that the model had big improvements regarding travel time reduction and congested route avoidance compared to typical shortest-path algorithms such as Dijkstra's and A* algorithms.

Jankovic (2021) proposed a multi-agent reinforcement learning approach to optimize route planning for a fleet of self-driving cars. Each vehicle was considered an agent in the research, and strategies were adopted for centralized training with decentralized execution. Empirical results demonstrated that the multi-agent reinforcement learning algorithm outperformed the conventional single-agent approach. Multi-agent RL also yielded a more balanced network-wide traffic flow.

Recently, Graph Neural Networks have been attracting attention for the route optimization problem in metropolitan areas because of their intrinsic graph structure of congestive flow. Gopi et al. (2023), proposed a GNN-based solution to dynamic route planning in AVs considering real-time conditions of flow and topology of road networks. These empirical results verify that GNN-based methods work better in route optimization compared with traditional graph-based algorithms, especially in reducing travel time and computation complexity.

3. Methodology

3.1 Dataset

This research project used the Road Traffic Flow Dataset from Kaggle which contained 48,000 records of traffic flow and included the following key features. This dataset is publicly available and designed for traffic prediction. The dataset is obtained from the respective traffic sensors, usually induction loops, and has been widely used for traffic flow forecasting and thus helpful for changing the settings of traffic light controllers (Pro-AI-Rokibul, 2024). It includes data measured over 56 days from six urban intersections, and the data is given as a flow time series showing the number of vehicles passing every five minutes throughout the day. The dataset is thus suitable for short-term forecasts. In this paper, data from four of those six intersections are used to simulate the four lanes of an intersection.

3.2 Data Pre-processing

This data pre-processing essentially focused on the cleaning of the data collected and making such data ready for modeling. Some of the key activities included handling missing values, removal of outliers in the dataset, normalization of numerical features, and encoding of categorical variables. Feature selection techniques were also employed to identify the most relevant attributes that relate closely to house price predictions (Pro-AI-Rokibul, 2024). Then stratified sampling techniques of data splitting were used on

the cleaned information to divide it into training and test data sets, which later proved helpful in the fitting models while objectively assessing their predictive accuracy.

3.3 Feature Engineering

For the dataset on traffic flow, there were a host of features, therefore, feature engineering entailed choosing only the relevant features to train the model. This comprised using techniques such as forward selection or backward elimination. Lastly, the dataset collected was appropriately encoded so it might be usable in the model training. One such was the application of one-hot encoding. The technique appends a new binary column to the list of unique categories of each variable.

Feature	Description
Holiday	Refers to if the day is a holiday.
temp	Temperature in Kelvin.
Rain_1h	The volume of rainfall in the last 1 hour [in mm].
Snow_1h	The volume of snowfall in the previous hour [in mm].
Cloud_ all	Percentage of cloud coverage.
Weather_ main	General weather conditions [clouds, clear, rain]
Date_ time	Timestamp of the record
Traffic_ volume	The target variable denotes the number of cars observed at the station.

3.4 Algorithms Deployed

In our study, renowned and proven machine learning models, particularly, linear regression and random forest were deployed. These algorithms were separately trained by using a part of the preprocessed data; then the same trained models were tested on an independent dataset concerning their predictive accuracy. This therefore gave us the avenue to compare the different modeling approaches and choose the best-suiting solution for our house price forecasting problem. This included a set of pre-processing and test activities: hyperparameter tuning for the algorithms with various methods to optimize the performance, including cross-validation and grid search. It represents a fair comparison and evaluation of predictive capability and accuracy amongst the different models usually trained by computing various metrics like mean square error, root mean squared error, and R-squared.

3.4.1 Linear Regression

As per Kumar & Saha (2020), Linear regression is considered a form of supervised machine learning used to make predictions from one or more input variables on an output variable. It defines a relation to the dependent variable from the input variables by which the method fits a linear equation to the collected data. In other words, there would be a line if there is one variable input, and in the case of having more than one variable input, a hyperplane would be cast, which tries to minimize the sum of the squared differences between the predicted and actual outcomes. The model computes an estimated coefficient for each input feature, hereby capturing the linear relationships in the data. Linear regression is a staple in any developer's toolkit, serving both as a simple means to understand and interpret the relationship between two variables and thus forming the basis of predictive analytics where appropriate.

3.4.2 Random Forest

Hernández-Mejía (2022), indicates that Random Forest is an ensemble learning method, useful both for classification and regression problems. While it creates as many decision trees in the process of training, it combines the outputs to ensure improved accuracy with a minimum risk of overfitting. Every tree is grown on a random subset of data and features, which further increases the diversity among the trees. For regression tasks, the final prediction is done with the average outcome, whereas in classification problems, a majority vote is used. This reduces variance and can improve the generalization capability of the model for complex datasets, adding robustness and performance when faced with noisy or missing values.

3.5 Experimentation Result

In Python, we deployed two regression models from the Scikit-learn library, most notably, Linear Regression and Random Forest Regressor, all of them at a random state equal to zero and default parameters for the reproducibility of the experiment.

3.6 Importing Libraries

```
import numpy as np
import seaborn as sns
import pandas as pd
from matplotlib import pyplot as plt
import warnings

# Ignore all warnings
warnings.filterwarnings("ignore")
```

```
df = pd.read_csv("Metro_Interstate_Traffic_Volume.csv")
df
```

Output:

	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description	date_time	traffic_volume
0	NaN	288.28	0.0	0.0	40	Clouds	scattered clouds	2012-10-02 09:00:00	5545
1	NaN	289.36	0.0	0.0	75	Clouds	broken clouds	2012-10-02 10:00:00	4516
2	NaN	289.58	0.0	0.0	90	Clouds	overcast clouds	2012-10-02 11:00:00	4767
3	NaN	290.13	0.0	0.0	90	Clouds	overcast clouds	2012-10-02 12:00:00	5026
4	NaN	291.14	0.0	0.0	75	Clouds	broken clouds	2012-10-02 13:00:00	4918
...	...	...	...	...	...	...	...	...	...
48199	NaN	283.45	0.0	0.0	75	Clouds	broken clouds	2018-09-30 19:00:00	3543
48200	NaN	282.76	0.0	0.0	90	Clouds	overcast clouds	2018-09-30 20:00:00	2781
48201	NaN	282.73	0.0	0.0	90	Thunderstorm	proximity thunderstorm	2018-09-30 21:00:00	2159
48202	NaN	282.09	0.0	0.0	90	Clouds	overcast clouds	2018-09-30 22:00:00	1450
48203	NaN	282.12	0.0	0.0	90	Clouds	overcast clouds	2018-09-30 23:00:00	954

To obtain the dataset overview, the analyst performed respective code snippets to obtain the following outcome:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48204 entries, 0 to 48203
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   holiday              61 non-null     object
 1   temp                 48204 non-null  float64
 2   rain_1h              48204 non-null  float64
 3   snow_1h              48204 non-null  float64
 4   clouds_all           48204 non-null  int64
 5   weather_main         48204 non-null  object
 6   weather_description  48204 non-null  object
 7   date_time            48204 non-null  object
 8   traffic_volume       48204 non-null  int64
dtypes: float64(3), int64(2), object(4)
memory usage: 3.3+ MB
```

```
df.describe()
```

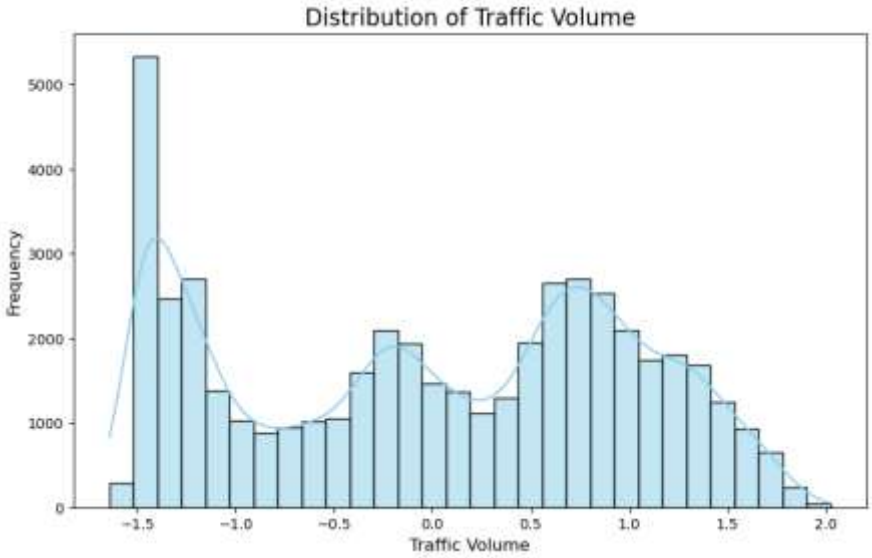
Output:

	temp	rain_1h	snow_1h	clouds_all	traffic_volume
count	48204.000000	48204.000000	48204.000000	48204.000000	48204.000000
mean	281.205870	0.334264	0.000222	49.362231	3259.818355
std	13.338232	44.789133	0.008168	39.015750	1986.860670
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	272.160000	0.000000	0.000000	1.000000	1193.000000
50%	282.450000	0.000000	0.000000	64.000000	3380.000000
75%	291.806000	0.000000	0.000000	90.000000	4933.000000
max	310.070000	9831.300000	0.510000	100.000000	7280.000000

To ascertain the traffic flow distribution, an exploratory data analysis was conducted to generate an appropriate histogram as follows:

```
# 1. Distribution of 'traffic_volume'
plt.figure(figsize=(10,6))
sns.histplot(df['traffic_volume'], kde=True, bins=30, color='skyblue')
plt.title('Distribution of Traffic Volume', fontsize=16)
plt.xlabel('Traffic Volume', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.show()
```

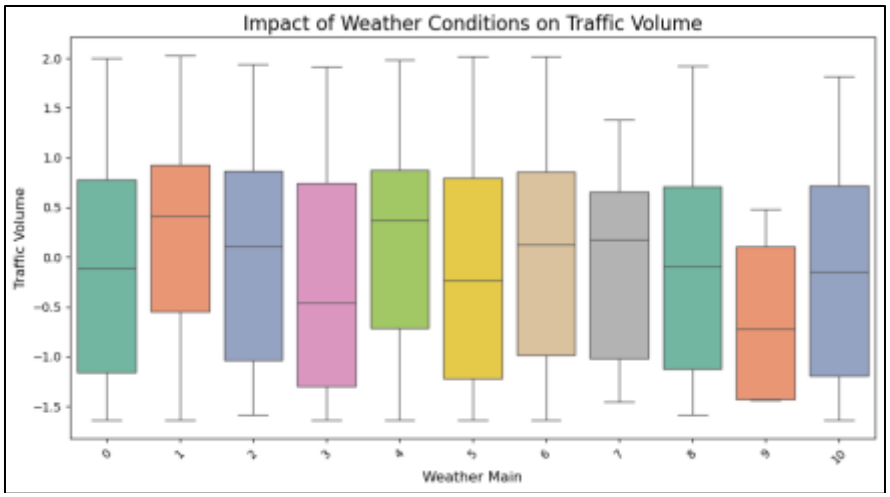
Output:



The analyst was also keen to pinpoint the impact of weather on traffic flow. By applying a suitable code snippet, the outcome was as displayed below:

```
# 4. Traffic volume by weather_main
plt.figure(figsize=(12,6))
sns.boxplot(x='weather_main', y='traffic_volume', data=df, palette='Set2')
plt.title('Impact of Weather Conditions on Traffic Volume', fontsize=16)
plt.xlabel('Weather Main', fontsize=12)
plt.ylabel('Traffic Volume', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```

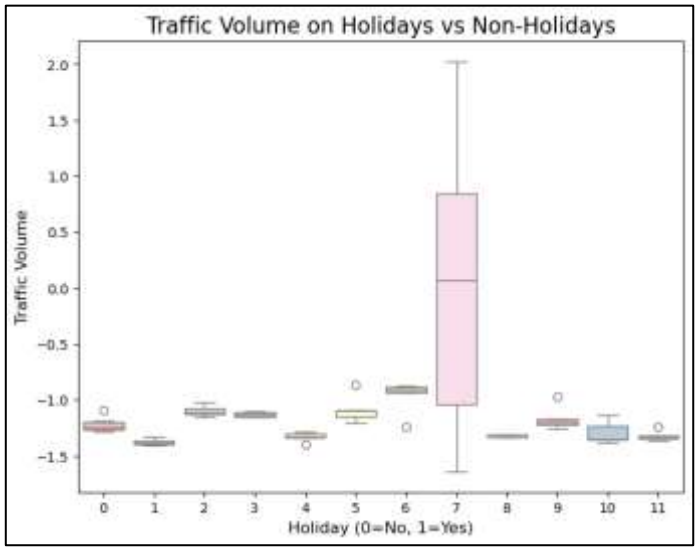
Output:



Moreover, to determine the traffic volume on holidays vs. non-holidays further exploratory analyses were undertaken culminating in the following results:

```
# 5. Holiday vs. Non-Holiday Traffic Volume
plt.figure(figsize=(8,6))
sns.boxplot(x='holiday', y='traffic_volume', data=df, palette='Pastel1')
plt.title('Traffic Volume on Holidays vs Non-Holidays', fontsize=16)
plt.xlabel('Holiday (0=No, 1=Yes)', fontsize=12)
plt.ylabel('Traffic Volume', fontsize=12)
plt.show()
```

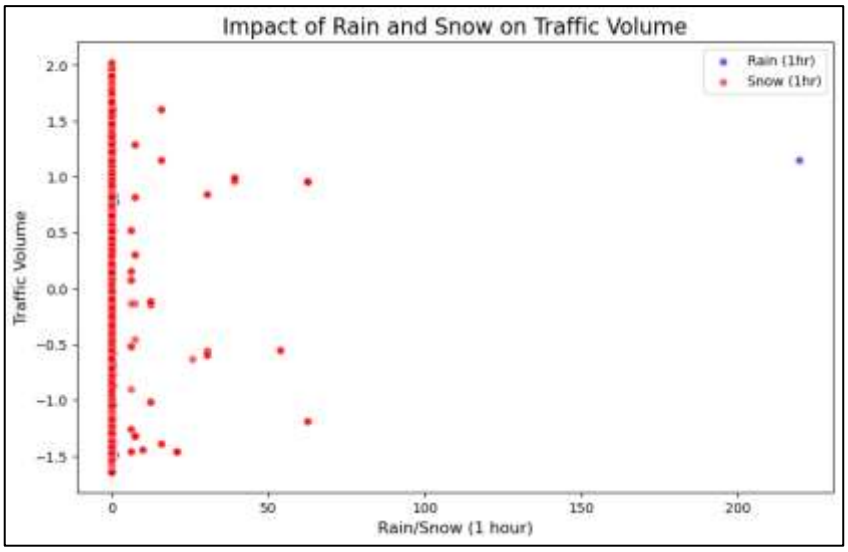
Output:



The researcher was also keen to assess the impact of rain and snow on traffic volume in further exploratory analysis as exhibited below:

```
# 6. Rain and snow impact on traffic volume
plt.figure(figsize=(10,6))
sns.scatterplot(x='rain_1h', y='traffic_volume', data=df, label='Rain (1hr)',
color='blue', alpha=0.6)
sns.scatterplot(x='snow_1h', y='traffic_volume', data=df, label='Snow (1hr)',
color='red', alpha=0.6)
plt.title('Impact of Rain and Snow on Traffic Volume', fontsize=16)
plt.xlabel('Rain/Snow (1 hour)', fontsize=12)
plt.ylabel('Traffic Volume', fontsize=12)
plt.legend()
plt.show()
```

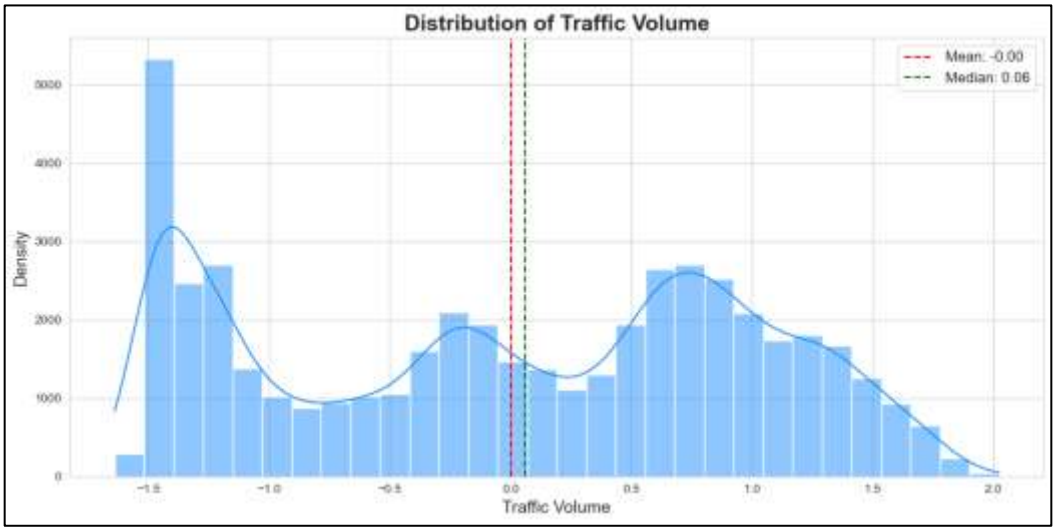
Output:



To establish the traffic volume distribution further code snippets were computed and the outcomes are presented below:

```
# Set a custom aesthetic for plots
sns.set_style("whitegrid")
sns.set_palette("Set2")

# 1. Distribution of 'traffic_volume' with insights on peak times
plt.figure(figsize=(12,6))
sns.histplot(df['traffic_volume'], kde=True, bins=30, color='dodgerblue')
plt.title('Distribution of Traffic Volume', fontsize=18, fontweight='bold')
plt.xlabel('Traffic Volume', fontsize=14)
plt.ylabel('Density', fontsize=14)
plt.axvline(df['traffic_volume'].mean(), color='red', linestyle='--', label=f'Mean:
{df["traffic_volume"].mean():.2f}')
plt.axvline(df['traffic_volume'].median(), color='green', linestyle='--',
label=f'Median: {df["traffic_volume"].median():.2f}')
plt.legend(fontsize=12)
plt.tight_layout()
plt.show()
```

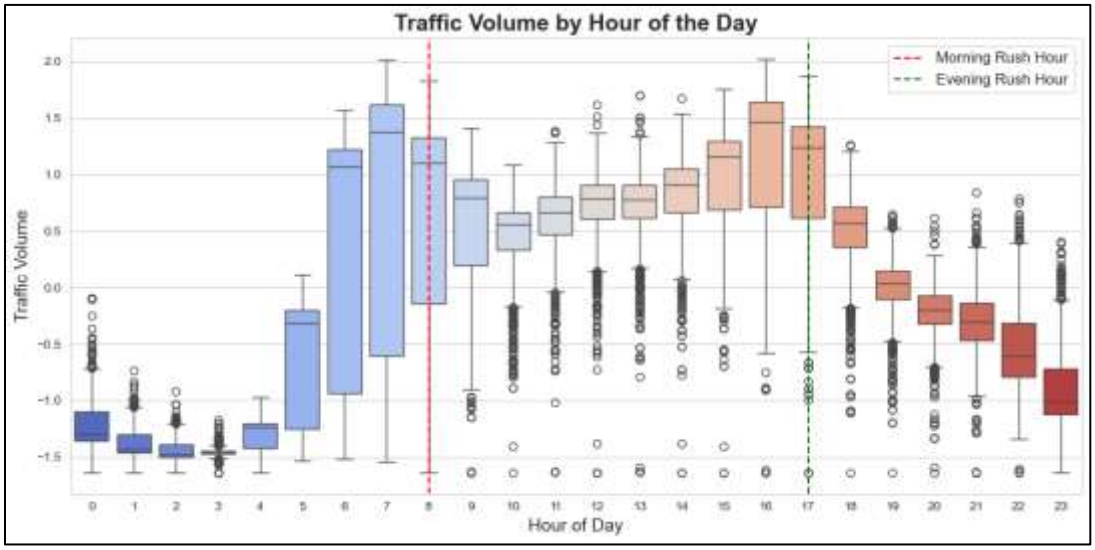
Output:



To get insights about traffic volume by hour of the day, further exploratory analyses were performed generating the following results.

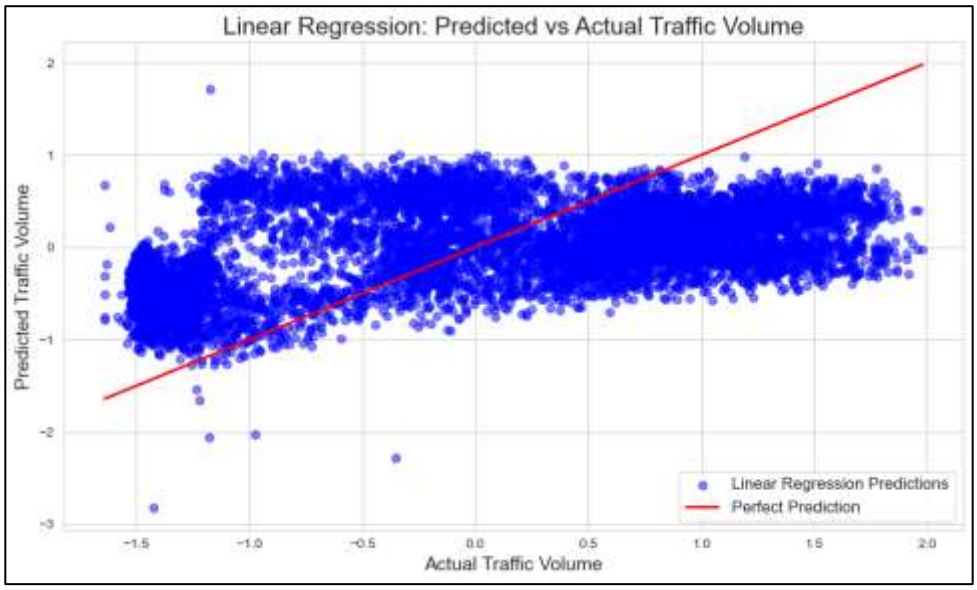
```
# Insight: Features like temperature and cloud cover have a slight correlation with traffic volume, suggesting weather patterns play a role.

# 3. Traffic volume by hour of the day with rush hour highlights
plt.figure(figsize=(12,6))
sns.boxplot(x='hour', y='traffic_volume', data=df, palette='coolwarm')
plt.title('Traffic Volume by Hour of the Day', fontsize=18, fontweight='bold')
plt.xlabel('Hour of Day', fontsize=14)
plt.ylabel('Traffic Volume', fontsize=14)
plt.axvline(x=8, color='red', linestyle='--', label='Morning Rush Hour')
plt.axvline(x=17, color='green', linestyle='--', label='Evening Rush Hour')
plt.legend(fontsize=12)
plt.tight_layout()
plt.show()
```

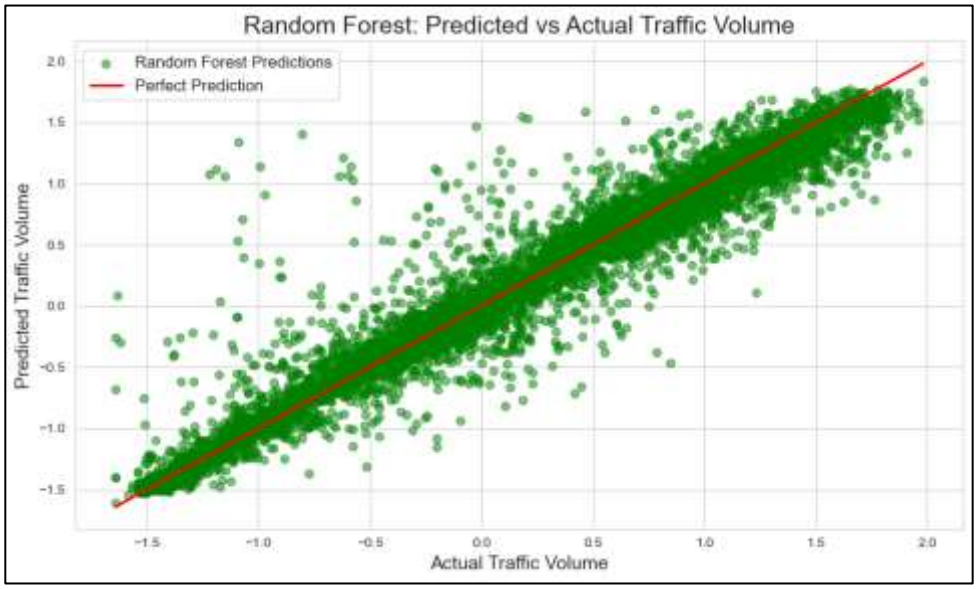
Output:



Model Performance Metrics

Model	Mean Squared Error (MSE)	R^2 Score
Linear Regression	0.80	0.20
Random Forest Regressors	0.04	0.96





3.7 Model Performance Summary

The MSE of the Linear Regression model was 0.80, which meant that the average of squared differences of predicted traffic volume against ground truth actual traffic volume. A lower MSE implied fewer large errors in predictions. Considering this outcome, the error was relatively high compared to the Random Forest model, which implies that the Linear Regression model cannot predict well. In particular, the MSE for the Random Forest model was significantly lower at 0.04, implying that the Random Forest Regressor had much smaller errors in predicting the volume of traffic, compared to the Linear Regression model. The lower the value of MSE, the better the model performance regarding the minimization of huge prediction errors.

As regards the R2 Score for the linear regression, the value was 0.20. This result meant that only 20% of the variance of the target variable was explained by the features. Low R^2 indicates improper fitness of the model to the data to capture the relationships between features and targets. The R^2 score of the Random Forest Model was relatively high at 0.96, proving that the explained

variance in traffic volume by this model was 96%. This outcome implied that the Random Forest regressors model excellently fitted the data, capturing most of the important patterns and relationships between the input features and target variable.

4. Conclusion

Autonomous vehicles represent one of the major revolutions that contemporary transportation has undergone. This research paper explored the AI and Machine Learning methodologies in traffic prediction and route optimization, and implications for autonomous vehicles in urban mobility. This research project used a road traffic flow Dataset from Kaggle which contained 48,000 records of traffic flow and included the following key features. In our study, renowned and proven machine learning models, particularly, linear regression and random forest were deployed. These algorithms were separately trained by using a part of the preprocessed data. The MSE for the Random Forest model was significantly lower, implying that the Random Forest Regressor had much smaller errors in predicting the volume of traffic, compared to the Linear Regression model. The R^2 score of the Random Forest Model was relatively high, proving that the explained variance in traffic volume by this model was commendable. This outcome implied that the Random Forest regressors model excellently fitted the data, capturing most of the important patterns and relationships between the input features and target variable.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Ahmad, M., Ali, M. A., Hasan, M. R., Mobo, F. D., & Rai, S. I. (2024). Geospatial Machine Learning and the Power of Python Programming: Libraries, Tools, Applications, and Plugins. In *Ethics, Machine Learning, and Python in Geospatial Analysis* (223-253). IGI Global.

[2] Hernández-Mejía, C. (2022). Traffic flow prediction for smart traffic lights using machine learning algorithms. *Uv-mx*. https://www.academia.edu/111084138/Traffic_Flow_Prediction_for_Smart_Traffic_Lights_Using_Machine_Learning_Algorithms?b=hybrid

[3] Hasan, M. R., Shawon, R. E. R., Rahman, A., Al Mukaddim, A., Khan, M. A., Hider, M. A., & Zeeshan, M. A. F. (2024). Optimizing Sustainable Supply Chains: Integrating Environmental Concerns and Carbon Footprint Reduction through AI-Enhanced Decision-Making in the USA. *Journal of Economics, Finance and Accounting Studies*, *6*(4), 57-71.

[4] Gopi, J., Vamshi, M., & Srinkath, K. (2023). Traffic prediction for intelligent transportation systems using machine learning. *Technoscienceacademy*. https://www.academia.edu/101285588/Traffic_Prediction_for_Intelligent_Transportation_Systems_Using_Machine_Learning?b=hybrid

[5] Jankovic, S. (2021). Traffic flow prediction using Machine learning. *www.academia.edu*. https://www.academia.edu/95645764/Traffic_Flow_Prediction_Using_Machine_Learning?b=hybrid

[6] Khan, M. A., Debnath, P., Al Sayeed, A., Sumon, M. F. I., Rahman, A., Khan, M. T., & Pant, L. (2024). Explainable AI and Machine Learning Model for California House Price Predictions: Intelligent Model for Homebuyers and Policymakers. *Journal of Business and Management Studies*, *6*(5), 73-84.

[7] Kumar, S., Patel, A., & Shankar, L. (2022). Traffic flow prediction using machine learning algorithms. *Irjet*. https://www.academia.edu/86021112/Traffic_Flow_Prediction_Using_Machine_Learning_Algorithms?b=hybrid

[8] Kumar, A., & Saha, P. (2020, December). A review of deep learning models for traffic flow prediction in Autonomous Vehicles. In 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN) (pp. 303-308). IEEE.

[9] Miglani, A., & Kumar, N. (2019). Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges. *Vehicular Communications*, *20*, 100184.

[10] Mushtaq, A., Haq, I. U., Imtiaz, M. U., Khan, A., & Shafiq, O. (2021). Traffic flow management of autonomous vehicles using deep reinforcement learning and smart rerouting. *IEEE Access*, *9*, 51005-51019.

[11] Navarro-Espinoza, A., López-Bonilla, O. R., García-Guerrero, E. E., Tlelo-Cuautle, E., López-Mancilla, D., Hernández-Mejía, C., & Inzunza-González, E. (2022). Traffic flow prediction for smart traffic lights using machine learning algorithms. Technologies, 10(1), 5.

[12] Pro-AI-Rokibul. (2024). *Traffic-Flow-Price-Predicton/Model/main.ipynb at main · proAIrokibul/Traffic-Flow-Price-Predicton*. GitHub. https://github.com/proAIrokibul/Traffic-Flow-Price-Predicton/blob/main/Model/main.ipynb

[13] Ramchandra, N. R., & Rajabhushanam, C. (2022). Machine learning algorithms performance evaluation in traffic flow prediction. Materials Today: Proceedings, 51, 1046-1050.