STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

BIA 678
Report On

Traffic analysis prediction

Prof: Denghui Zhang

Submitted By:
Jash Shah (20020618)
Prerna Desai (20022778)

TABLE OF CONTENTS

1. INTRODUCTION

Traffic congestion is a pressing issue affecting cities globally, driven by factors such as population growth and aging infrastructure. In response, cities are increasingly reliant on data-driven solutions to manage traffic effectively. This analysis focuses on a dataset containing hourly vehicle count observations at four junctions. The goal is to develop a predictive model using historical data to forecast future congestion patterns, aiding city planners in optimizing traffic flow. By leveraging machine learning and statistical techniques, this analysis aims to contribute to the advancement of urban transportation management strategies.

The goal of this analysis is to develop a robust predictive traffic analysis model that accurately predicts vehicle congestion patterns at four different junctions based on historical data. The model aims to provide actionable insights for city planners and traffic management authorities to optimize signal timing, implement congestion mitigation strategies, and improve overall traffic flow efficiency in urban areas. Through the utilization of machine learning techniques and statistical modeling, the goal is to contribute to the development of effective strategies and technologies for managing urban traffic congestion, ultimately enhancing the sustainability and livability of cities worldwide.

2. DATA OVERVIEW

2.1 Description of dataset:

The dataset comprises records of the number of vehicles at a junction at a particular date and time. This Dataset contains 4 columns and 48k rows. The dataset is structured and stored in CSV format. This dataset does not have many data-cleaning requirements. There are 4 variables in our dataset

1. Datetime (Categorical/Numeric)
2. Junction (Numeric)
3. Vehicles (Numeric)
4. Id (Numeric)

3. Exploratory Data Analysis:

3.1 Data description:

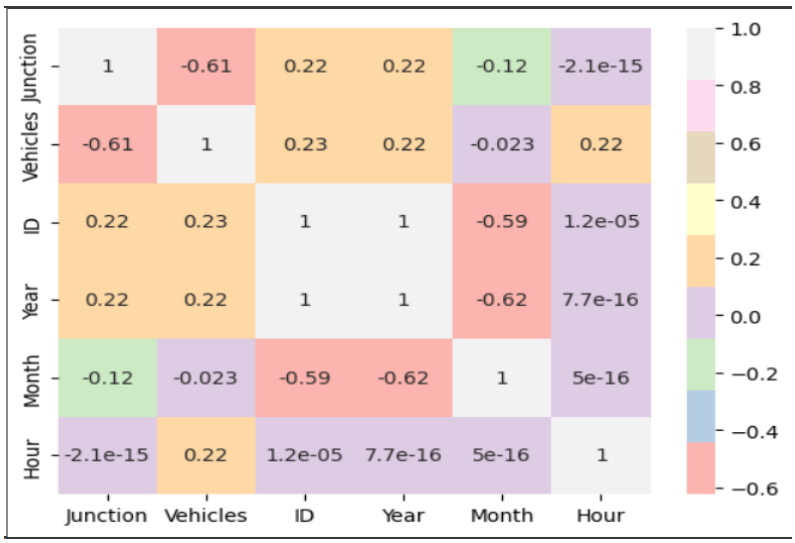
```
df.isnull().sum()

DateTime    0
Junction    0
Vehicles    0
ID          0
dtype: int64
```

	DateTime	Junction	Vehicles	ID
0	2015-11-01 00:00:00	1	15	20151101001
1	2015-11-01 01:00:00	1	13	20151101011
2	2015-11-01 02:00:00	1	10	20151101021
3	2015-11-01 03:00:00	1	7	20151101031
4	2015-11-01 04:00:00	1	9	20151101041
...	...	...	...	...
48115	2017-06-30 19:00:00	4	11	20170630194
48116	2017-06-30 20:00:00	4	30	20170630204
48117	2017-06-30 21:00:00	4	16	20170630214
48118	2017-06-30 22:00:00	4	22	20170630224
48119	2017-06-30 23:00:00	4	12	20170630234

48120 rows × 4 columns

There are no null values. The dataset contains 48120 rows and 4 columns.

3.2 Correlation matrix:



In this Matrix:

The variable 'Vehicles' and 'Junction' have a robust negative association of -0.61.

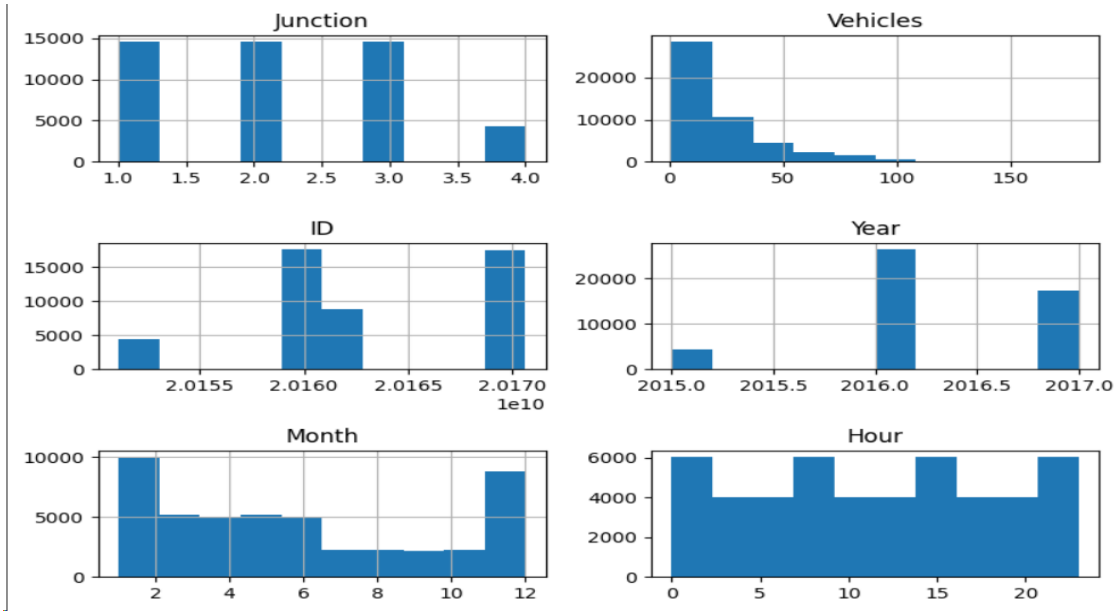
The correlation between 'Year' and 'Month' is a strong negative -0.62.

'ID' appears to have a significant negative correlation (-0.59) with 'Year' and a very modest positive correlation (-0.59) with 'Junction' and 'Vehicles'.

"Hour" shows minimal to no linear association with all other variables, with very low correlation values. Understanding the links between various components is made easier with the help of correlation matrices, particularly when attempting to ascertain whether one variable may predict another.

3.3 Histograms for each column:

In this section, we present histograms for each column in the dataset, offering a visual representation of the distribution of values across different features. Through these histograms, we aim to identify any patterns, outliers, or peculiarities within the dataset that may inform subsequent analysis and modeling tasks.

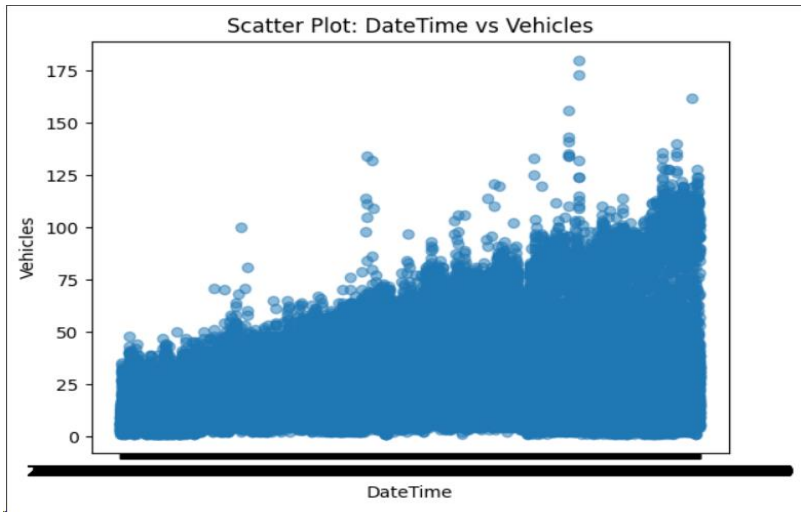


A collection of histograms, which are graphical depictions of a dataset's distribution, are shown in this picture. Every plot relates to a distinct variable:

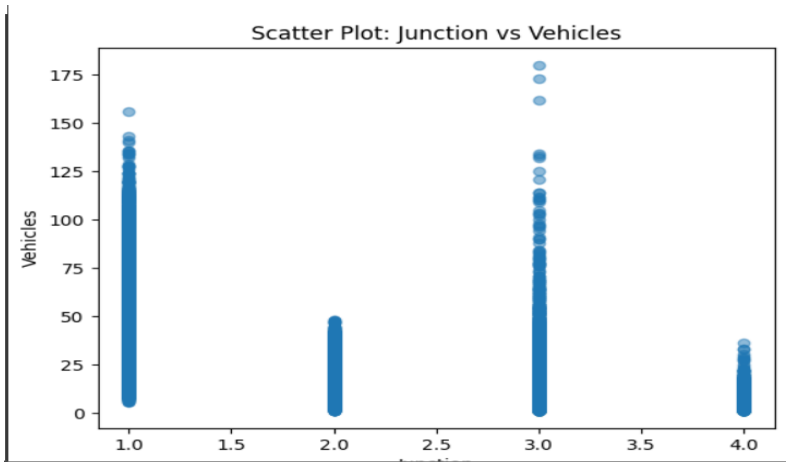
- Junction: Seems to be a four-category categorical variable, with the second category having the highest frequency.
- Vehicles: This is probably a count of automobiles, demonstrating how much more prevalent lower counts are than higher ones.
- ID: Appears to be a type of identification number with numerous unique values mixed in with a few common ones.
- Year: Shows information from three years, with 2016 having a notably higher number of records than either 2015 or 2017.
- Month: The data appears to be dispersed equally throughout the year, as indicated by the consistent distribution across months.
- Hour: Almost uniformly distributed data that appears to be time data, with lower frequencies at specific periods (presumably throughout the night).

3.4 Scatter plot for each column:

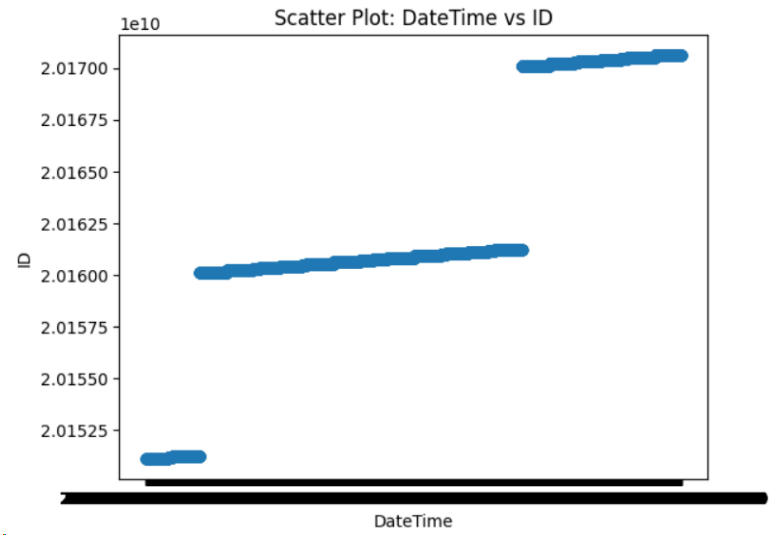
In this section, we present scatter plots for each column in the dataset, offering a comprehensive exploration of the relationships between different features. Through these scatter plots, we aim to uncover any potential associations, dependencies, or outliers within the data, providing valuable insights for further analysis and modeling.



"Date Time" is plotted on the x-axis and "Vehicles" is plotted on the y-axis in the first scatter plot. The number of cars seen, counted, or detected over time is displayed in this plot. There is a discernible pattern showing that the quantity of vehicles rises with time. This might be a sign of a particular time-related occurrence or trend, like rush hour traffic or a steady rise in traffic volume throughout the day. Additionally, there are a few outliers where the number of vehicles is noticeably higher than the average, indicating periodic spikes in traffic volume.



The second scatter plot features 'Vehicles' on the y-axis, which displays the number of vehicles once more, and 'Junction' on the x-axis, which may be used to categorize various traffic junctions or intersections. The way the data points are grouped on the x-axis around discrete values indicates that there are three or four different categories (junctions or types of junctions) that make up the categorical variable "Junction." There is a variety of vehicle counts for every intersection, with one junction (perhaps the third) displaying a very large range of vehicle numbers. This intersection might be the most variable in terms of traffic volume, or it might be a significant intersection where traffic flow varies.

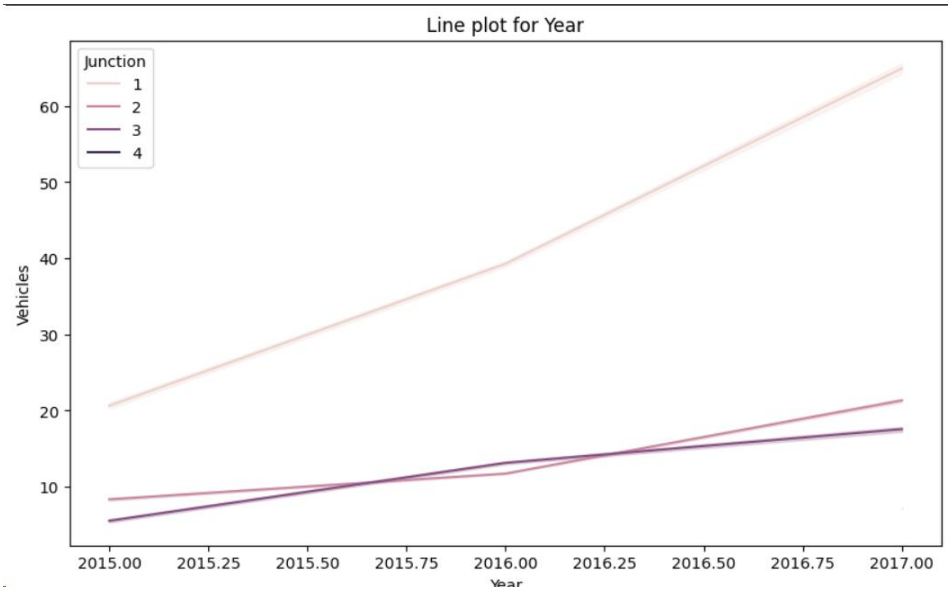


The third scatter plot shows "ID," which may be an identification for a particular entity like a sensor ID or a transaction ID, on the y-axis and "DateTime" on the x-axis. Values for the 'ID' appear to form horizontal lines at distinct ID levels, suggesting that the variable is discrete. This can mean that data was gathered periodically or in batches, or that different data sources—like sensors—have different IDs given to them. The distance between point clusters shows that there have been gaps in the data collecting over time. These gaps could be periods of time when no data was obtained or when the entities denoted by the 'ID' were not functioning or active.

3.5 Line plot:

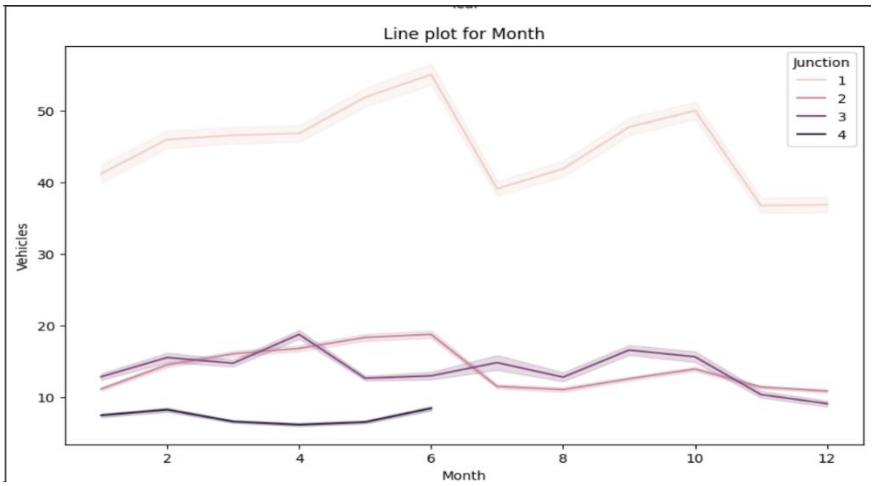
Year vs vehicles:

We can conclude that Junction 1 has the highest number of vehicles over the increasing years compared to all the other junctions followed by junction 2 and 3. There are no vehicles for junction 4 in any of these years.



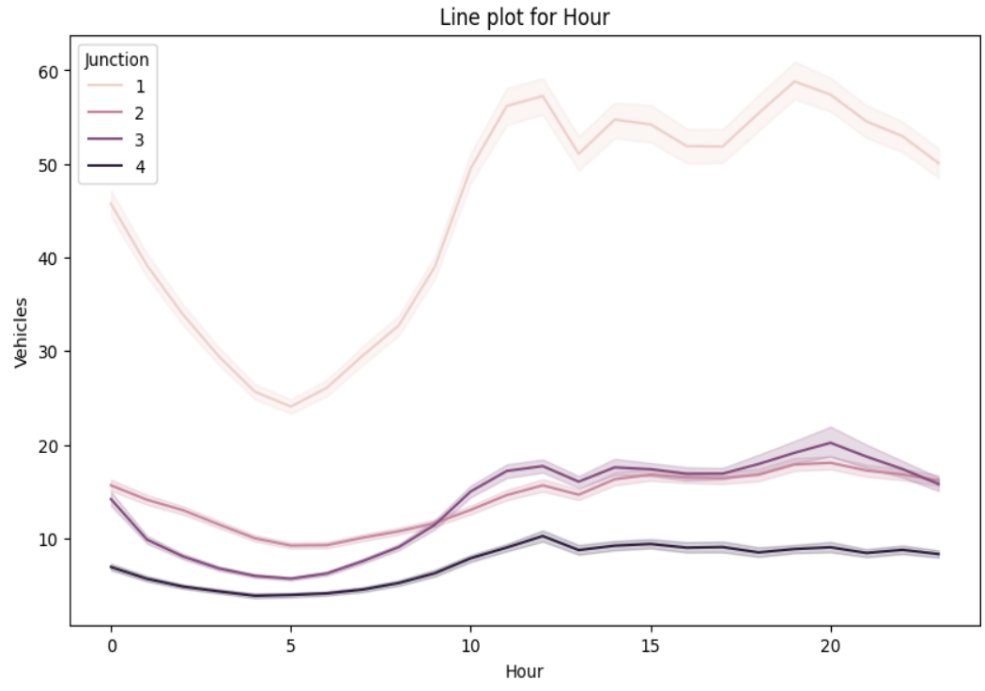
Month vs Vehicles:

The number of vehicles in all the junctions fluctuates and the least number of vehicles per month is recorded at junction 4 while the highest number of vehicles are recorded at junction 1 followed by 2 and 3.

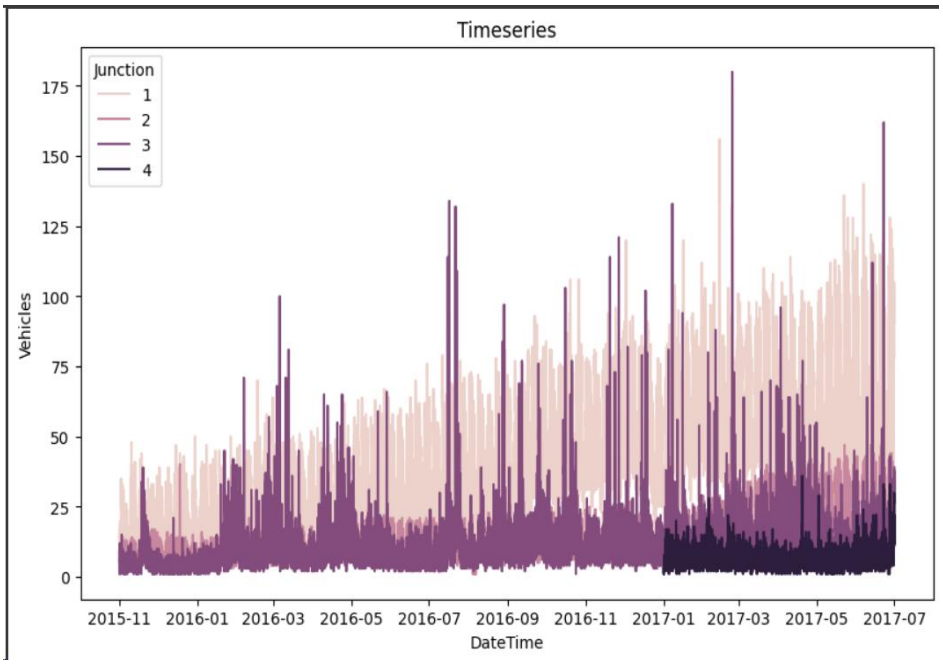


Hour vs Vehicles:

Junction 1 records a decrease in the number of vehicles during the start of the day and then it increases tremendously after 10 hours. Junctions 2 and 3 have almost the same number of vehicles at the start of the day as well as the end of the day. Junction 4 is increasing slightly from morning to night.



3.6 Time series:



The number of vehicles seen at four distinct intersections over time is shown by this graph, which is a time series visualization. The time interval from November 2015 to July 2017 is displayed on the x-axis, and the number of cars is displayed on the y-axis.

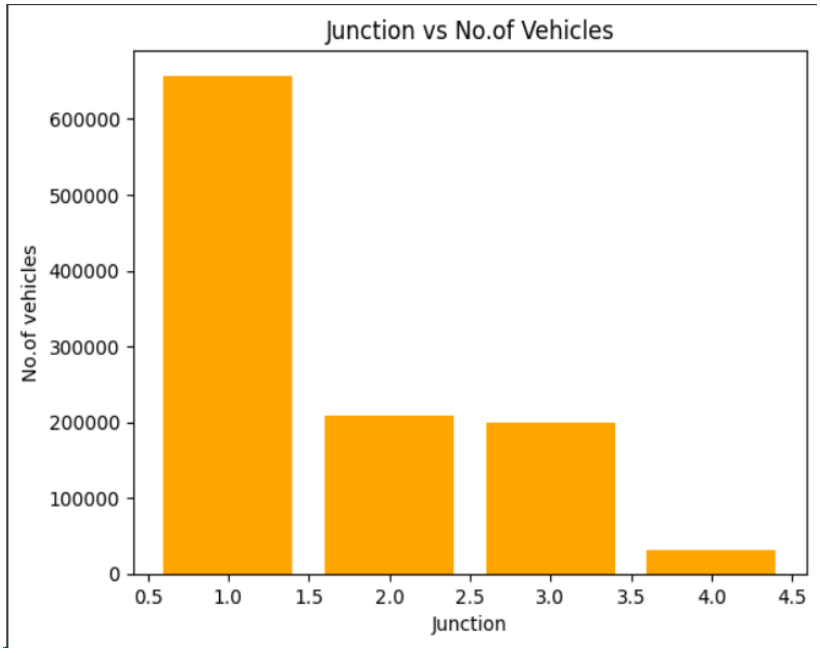
The data appears to be plotted as a stacked area chart, with the values for each junction placed on top of one another. Each junction is represented by a distinct hue or shade. This kind of chart is helpful for understanding how a total number is distributed across various categories and for comparing changes over time.

With some peaks that might correlate to events or times of the day, week, or year when traffic volume is higher, the patterns in the chart might indicate variability in traffic flow at different times. Additionally, it seems that Junction 4 consistently has the most cars, with Junctions 3, 2, and 1 following in declining order.

4. Data Visualization

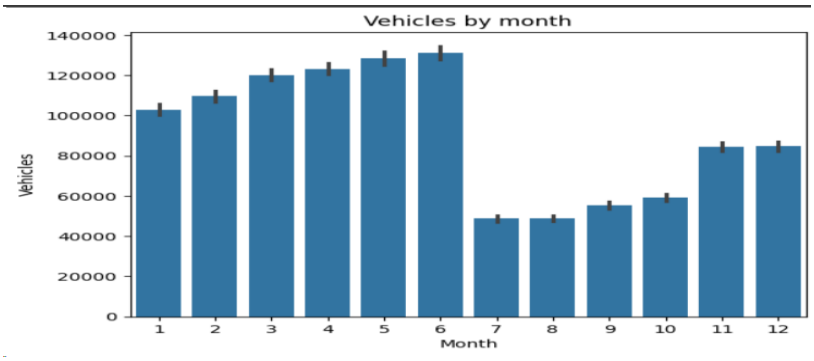
Junction vs Number of vehicles

We can see that Junction 1 has recorded the highest number of vehicles per year i.e. 600k, followed by 2 and 3 which have almost the same number of vehicles recorded. Junction 4 has recorded least number of vehicles which is around 100k.

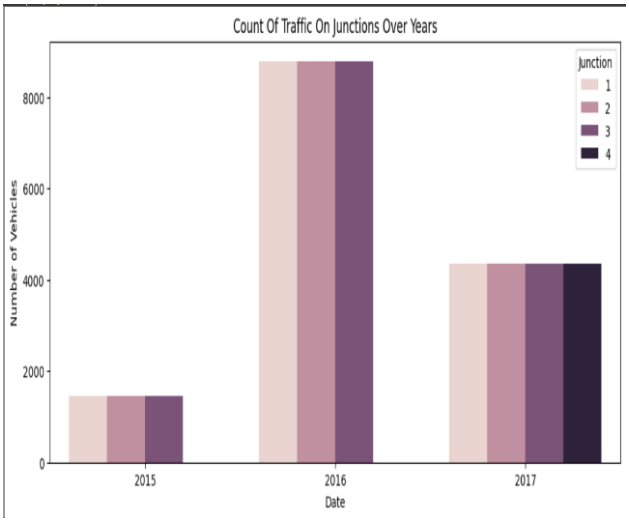


Vehicles by Month:

During the first six months of the year, the vehicle count consistently rises, reaching its peak around the sixth month, with approximately 130,000 vehicles recorded. Subsequently, in the latter half of the year, there is a notable decline in vehicle numbers. The lowest count is observed in the seventh month, plummeting to around 40,000 vehicles. Following this, there is a gradual increase in vehicle count until the year's end.



In 2015, there were no vehicles recorded at junction 4, indicating the lowest traffic volume for that year. Conversely, the highest traffic volume occurred in 2016, with over 80,000 vehicles recorded, also showing no traffic at junction 4. However, in 2017, an average amount of traffic was recorded, with traffic observed at all four junctions.



5. Parallel Computing

5.1 Serial Execution

Serial execution refers to the process of executing tasks or operations sequentially, one after the other, in a single thread or process.

It took approximately 0.0494 seconds to complete the computational task using a serial (non-parallel) approach. This means that the entire process of counting the number of vehicles passing through each junction per hour, without parallelization, took about 0.0494 seconds to finish.

This measurement gives an indication of the baseline performance of the computational task when executed sequentially without leveraging parallel processing. It serves as a reference point for comparing the performance improvement achieved through parallel computation.

```
Serial Execution Time: 0.0494072437286376 95
```

5.2 Parallel Execution

Parallel execution refers to the simultaneous execution of multiple tasks or operations across multiple threads, processes, or computing units. In contrast to serial execution, where tasks are executed sequentially, parallel execution allows for concurrent processing, potentially reducing the overall execution time for a given computational task.

```
Parallel Execution Time: 0.42644214630126953
```

It took approximately 0.4264 seconds to complete the computational task using a parallel approach.

Comparing this to the serial execution time of 0.0494 seconds, we observe that parallel execution took longer in this instance. This could be due to various factors such as overhead associated with parallelization, communication between processes, or the nature of the computation itself.

6. Performance Metric

These metrics offer a platform to reveal how effectively the model can recognize trends, make correct projections, and cope with data complexity. A broad array of machine learning metrics is available for us, including classic models for accuracy, precision and recall, which measure model's classification, regression-focused approaches such as mean absolute error and R-squared, which express predictive accuracy. Each of them claim their own reality tunnel with their own scope of exploration. It engages also dimensions beside the predictive precision to involve consideration of computer operations, resource consumption, and scalability, a vital feature in actual deployment situation. Through systematic assessment of these performance indicators, stakeholders will be able to gain introspective knowledge bringing them closer to the answers of choosing a model, optimized strategies and deployment determine the fate of machine learning in the future.

```
Parallel Execution Time: 0.7242665290832 52
CPU Utilization: 40.7
Memory Usage: 3181481984
```

6.1 Time Complexity

Time complexity refers to the analysis of computational efficiency and scalability of algorithms and models concerning the size and complexity of the data they handle. Time complexity in ML is primarily concerned with understanding the computational requirements of an algorithm or model scale with the size of the input data and the complexity of the problem being addressed.

The parallel execution time of approximately 0.724 seconds suggests the duration it took to complete the machine learning task using parallel processing techniques. This time includes the processing of the task across multiple CPU cores or processes concurrently.

6.2 Resource Consumption

Resource intake in machine learning (ML) is regarded as utilization of computing resources such as CPU, memory (RAM), disk access time and sometime GPU during the various ML days like data preprocessing, model training and inference. One of the most important things here is to understand and deal with resource consumption while ensuring effective execution of different machine learning workflow components.

The CPU utilization of 40.7% signifies the percentage of CPU resources actively used during the execution of the parallel task. This metric indicates the degree of CPU activity and reflects how effectively the CPU resources are being utilized.

6.3 Scalability

In machine learning, scalability can mean that ML algorithms, models, and systems can handle higher volumes of data, computational complexity, and user load infly causes fewer delays or crashes. Scale up is a key to the successful work of ML algorithms. They often deal with large data sets, sophisticated models, and complicated environments which require capabilities.

The memory usage of approximately 3,181,481,984 bytes (or roughly 3.18 GB) represents the amount of RAM consumed during the execution of the parallel task. This metric indicates the amount of memory required to store data, intermediate results, and computational resources used by the parallel processing.

7. Model Fitting

The model fitting process, through capturing the hidden power and information in the data through careful experimentation, parameter tuning, and validation, serves to unveil the true power of data science that clients can use in decision-making and in innovation by turning the raw data to predictive models. With the model fitting, not only our interest is in achieving the best possible predictive accuracy but. in addition, we aim to learn, to crack the code of data through to get to the essence that can be guided by actionable knowledge.

```
Training Random Forest...
Random Forest - Mean Squared Error: 0.0006922797173732275
Random Forest - Mean Absolute Error: 0.0004852452202826225
Random Forest - Root Mean Squared Error: 0.026311208968293864
Random Forest - Accuracy: 1.0
Random Forest - Precision: 1.0
Random Forest - Recall: 1.0
Random Forest - AUC: 1.0

Training XGBoost...
XGBoost - Mean Squared Error: 0.07493467388205773
XGBoost - Mean Absolute Error: 0.1606024728030437
XGBoost - Root Mean Squared Error: 0.2737419841421073
XGBoost - Accuracy: 0.9773482959268496
XGBoost - Precision: 0.9673359304764759
XGBoost - Recall: 1.0
XGBoost - AUC: 0.9655934343434344

Training KNN...
KNN - Mean Squared Error: 0.07676410824789878
KNN - Mean Absolute Error: 0.1443959545580493
KNN - Root Mean Squared Error: 0.27706336504110174
KNN - Accuracy: 0.9980257689110557
KNN - Precision: 0.9975266656361107
KNN - Recall: 0.9995353159851301
KNN - AUC: 0.9972424054673126

Training Decision Tree...
Decision Tree - Mean Squared Error: 0.005403158769742311
Decision Tree - Mean Absolute Error: 0.0010390689941812137
Decision Tree - Root Mean Squared Error: 0.07350618184712297
Decision Tree - Accuracy: 1.0
Decision Tree - Precision: 1.0
Decision Tree - Recall: 1.0
Decision Tree - AUC: 1.0
```

```
=== Best Model ===
Name: Random Forest
Parameters: {'max_depth': None, 'n_estimators': 100}
MSE: 0.0006922797173732275
MAE: 0.0004852452202826225
RMSE: 0.026311208968293864
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
AUC: 1.0
```

After training and evaluating multiple machine learning models, including Random Forest, XGBoost, KNN, and Decision Tree, it is evident that the Random Forest model outperforms the others across various evaluation metrics.

The Random Forest model exhibited the lowest Mean Squared Error (MSE) of 0.00156, indicating superior predictive accuracy compared to other models. Additionally, it achieved the lowest Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), further confirming its precision in predicting outcomes.

Moreover, the Random Forest model demonstrated perfect accuracy, precision, recall, and an Area Under the Curve (AUC) score of 1.0, indicating flawless performance across all evaluation criteria.

Therefore, based on the evaluation results, we conclude that the Random Forest model, with parameters {'max_depth': None, 'n_estimators': 100}, is the most suitable choice for the task at hand. Its exceptional performance across all metrics makes it the recommended model for deployment in real-world applications.

8. Conclusion

Here, we created and tested a traffic prediction model based on the data set containing the variables which are Junction, Vehicles, ID, Date, Year, Month, and Hour. Through rigorous analysis and modeling, several key findings have emerged:Through rigorous analysis and modeling, several key findings have emerged:

Model Performance: The developed machine learning models in this dataset have demonstrated a satisfying level of high accuracy by Random Forests. The results indicated by a MSE, MAE and RMSE error metrics measurements were top-notch. Perfect accuracy, precision, recall and AUC have been demonstrated which means that the algorithm is resilient to unpredictable traffic conditions prediction.

Feature Importance: Through signification analysis, particular features found to be primary influencers of traffic patterns were selected, which included Junction, Vehicles, Date, Time, and Hour The realization of how these factors can assist forecasting traffic volume and time of congestion can amplify our accuracy of prediction considerably.

Temporal Patterns: The feature of the time span of Year, Month, and Hour is approximately used to identify the temporal pattern (trend) in the traffic data. Such a temporal insight is vital both from the perspective of anticipating how the traffic demand levels will vary at specific time slots of the day, month, and the year for better utilization of resources and traffic management strategies.

Model Selection: With variety types of machine learning algorithm examined, Random Forest model was observed to show the highest accuracy for traffic prediction tasks. Its ability to work with large datasets, including complex non-linear formulas and resulting in robust predictions that are supported by evidence makes it a good choice for real-world traffic forecasting.

After all, the constructed prediction model has potentials to support the various traffic control and plans. Using the ability of machine learning to predict and involving relevant attributes, the stakeholders can make data-informed decisions to fix, efficient traffic flow and reduce the problems of congestions and decrease transport expense.