Industrial Internship Report on

”Forecasting of Smart City Traffic Patterns”

Prepared by

Adarsh Kumar Pandey

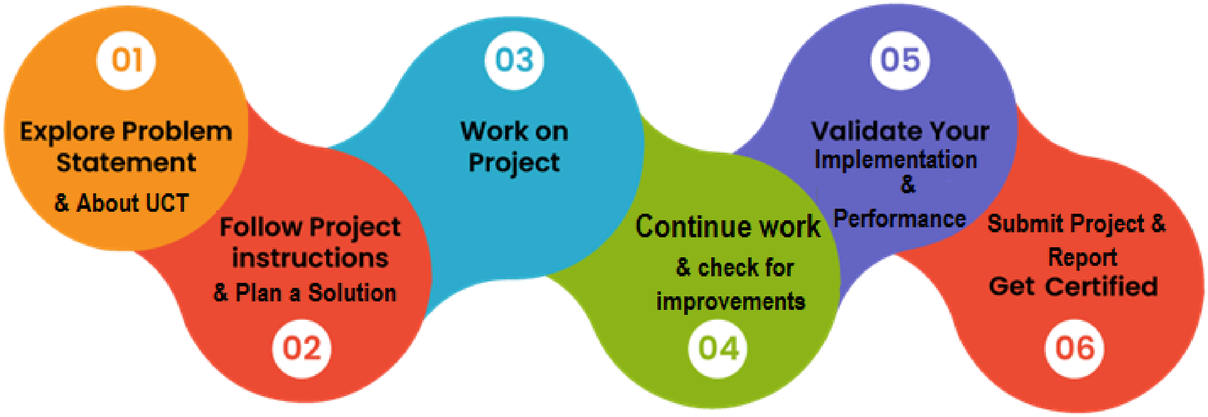
Executive Summary

My project was "Forecasting of Smart City Traffic Patterns". In this project, I developed a machine learning model to predict traffic volume at various city junctions. Using historical data, I performed data analysis, created new features based on time and holidays, and trained a Random Forest model. The final model achieved a high accuracy of 91.97% (R-squared) and was deployed as a web application using Streamlit.

TABLE OF CONTENTS

1 Preface

This six-week internship provided by upskill Campus and UniConverge Technologies Pvt Ltd was a crucial step in my career development. I was assigned the project "Forecasting of Smart City Traffic Patterns," which addresses the real-world challenge of urban traffic management. The program was well-structured, starting from understanding the problem to implementing and testing a machine learning solution.



This six-week internship provided by upskill Campus and UniConverge Technologies Pvt Ltd was a crucial step in my career development. I was assigned the project "Forecasting of Smart City Traffic Patterns," which addresses the real-world challenge of urban traffic management. The program was well-structured, starting from understanding the problem to implementing and testing a machine learning solution.

Throughout this journey, I gained hands-on experience in data analysis, model building, and web deployment. This opportunity has significantly improved my technical skills and problem-solving abilities. I am thankful to everyone at USC and UCT who guided me. This report documents the complete process and my learnings from this valuable experience.

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

uct

Uniconverge Technologies

IIOT Products

We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways

IIOT Solutions

We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence…

OEM Services

We offer solutions ranging from product design to final production we handle everything for you..

i. UCT IoT Platform (uct Insight)

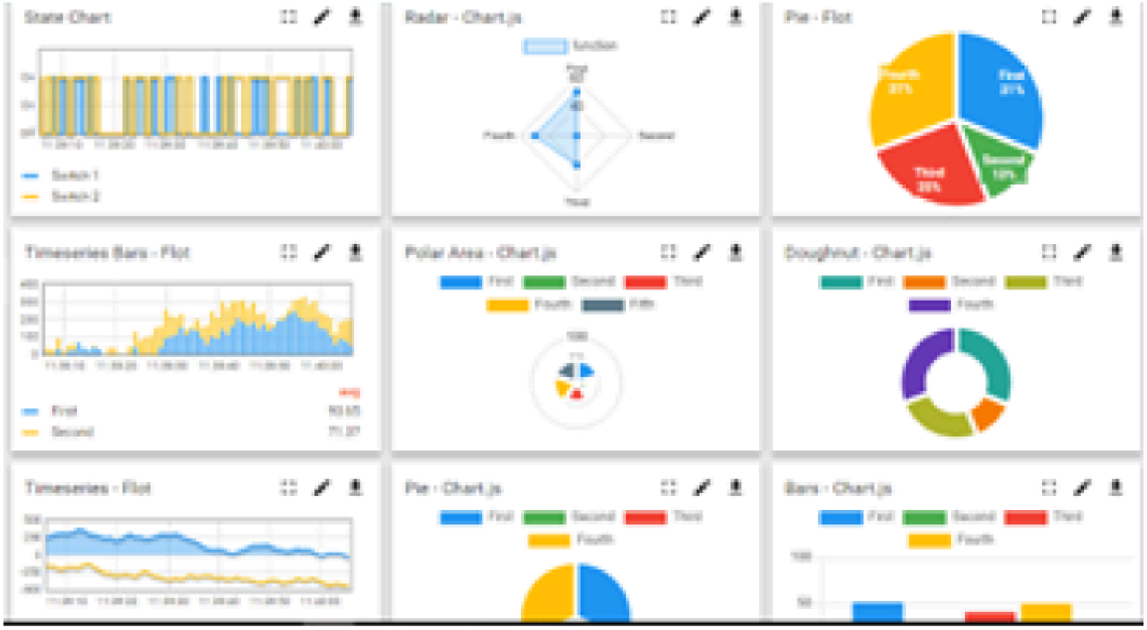
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

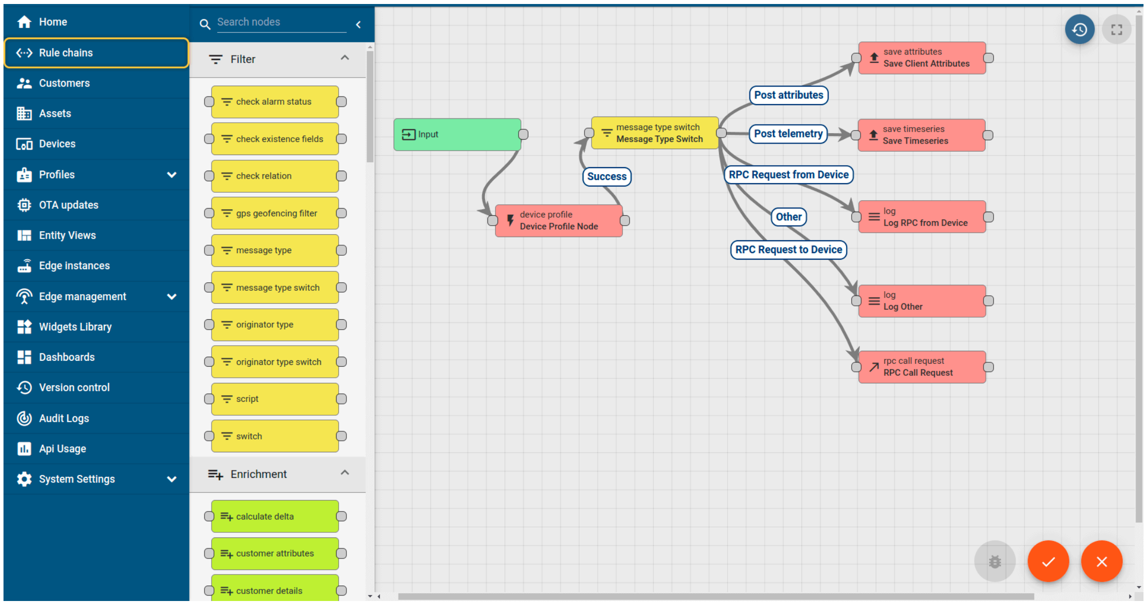
- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA

- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine


State Chart
Switch 1
Switch 2
Radar - Chart.js
Pie - Flot
First 31%
Second 13%
Third 25%
Fourth 31%
Timeseries Bars - Flot
First
Second
Polar Area - Chart.js
First
Second
Third
Fourth
Fifth
Doughnut - Chart.js
First
Second
Third
Fourth
Timeseries - Flot
Pie - Chart.js
First
Second
Third
Fourth
Bars - Chart.js
First
Second
Third
Fourth


Home
Rule chains
Customers
Assets
Devices
Profiles
OTA updates
Entity Views
Edge instances
Edge management
Widgets Library
Dashboards
Version control
Audit Logs
Api Usage
System Settings
Search nodes
Filter
check alarm status
check existence fields
check relation
gps geofencing filter
message type
message type switch
originator type
originator type switch
script
switch
Enrichment
calculate delta
customer attributes
customer details
Input
message type switch
Message Type Switch
device profile
Device Profile Node
Success
Post attributes
Post telemetry
RPC Request from Device
Other
RPC Request to Device
save attributes
Save Client Attributes
save timeseries
Save Timeseries
log
Log RPC from Device
log
Log Other
rpc call request
RPC Call Request

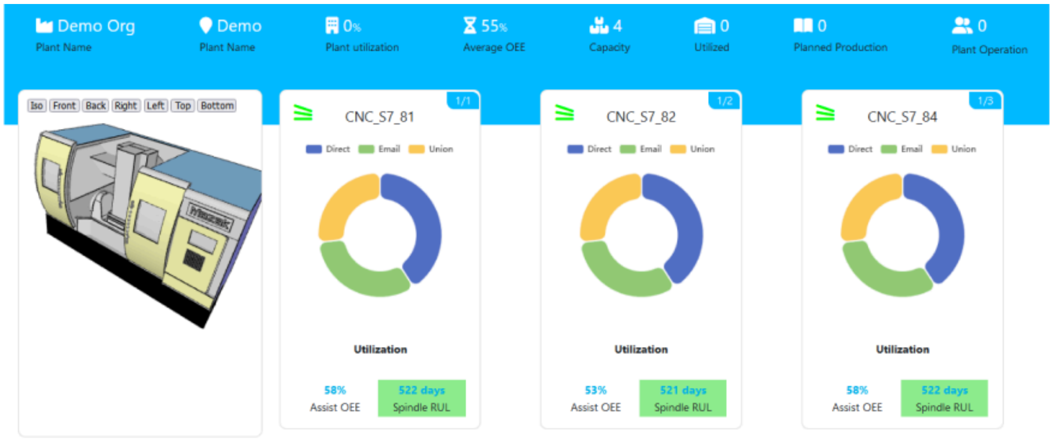
ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleash the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.



Demo Org
Plant Name

Demo
Plant Name

0%
Plant utilization

55%
Average OEE

4
Capacity

0
Utilized

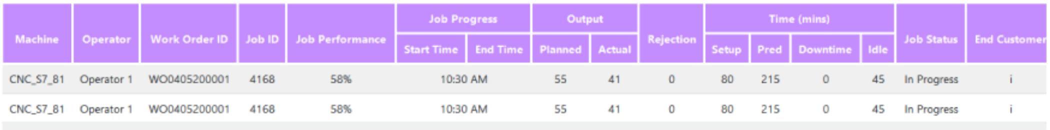
0
Planned Production

0
Plant Operation

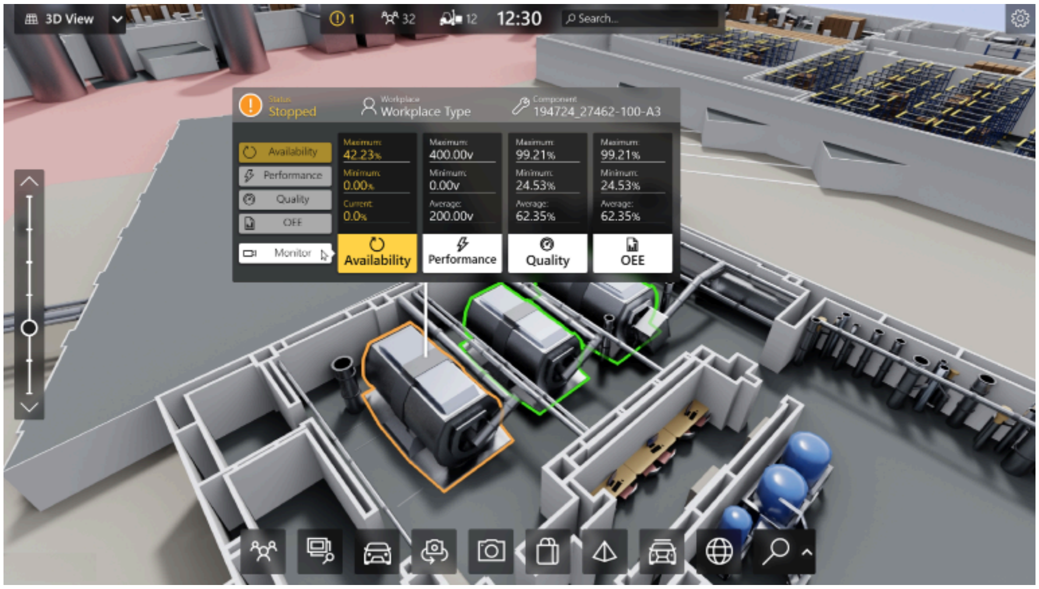
CNC_S7_81 1/1
Direct Email Union
Utilization
58% Assist OEE
522 days Spindle RUL

CNC_S7_82 1/2
Direct Email Union
Utilization
53% Assist OEE
521 days Spindle RUL

CNC_S7_84 1/3
Direct Email Union
Utilization
58% Assist OEE
522 days Spindle RUL



Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i



Status: Stopped
Workplace: Workplace Type
Component: 194724_27462-100-A3

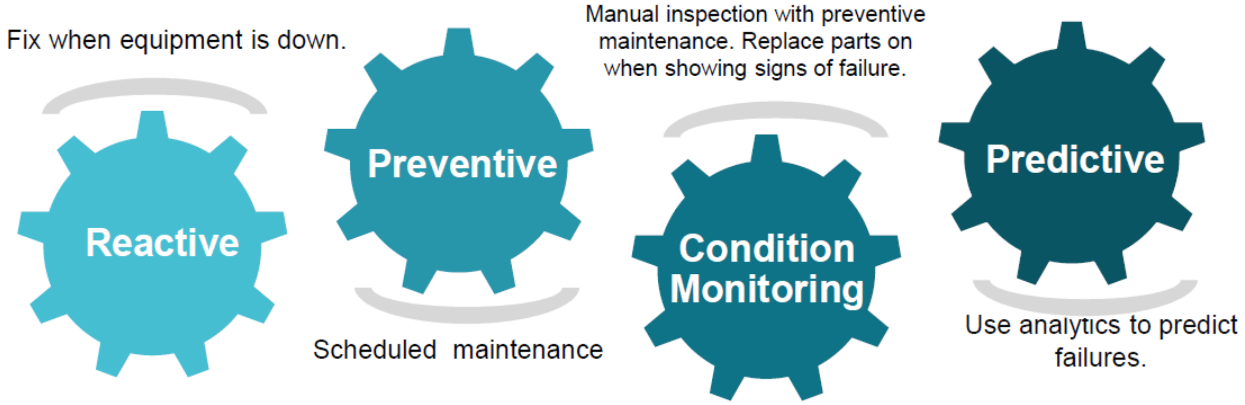
Availability — Maximum: 42.23% Minimum: 0.00% Current: 0.0%
Performance — Maximum: 400.00v Minimum: 0.00v Average: 200.00v
Quality — Maximum: 99.21% Minimum: 24.53% Average: 62.35%
OEE — Maximum: 99.21% Minimum: 24.53% Average: 62.35%

iii. LoRaWAN™ based Solution

UCT is one of the early adopters of LoRAWAN technology and provides solutions in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.


Fix when equipment is down.
Reactive
Preventive
Scheduled maintenance
Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.
Condition Monitoring
Predictive
Use analytics to predict failures.

2.2 About Upskill Campus (USC)

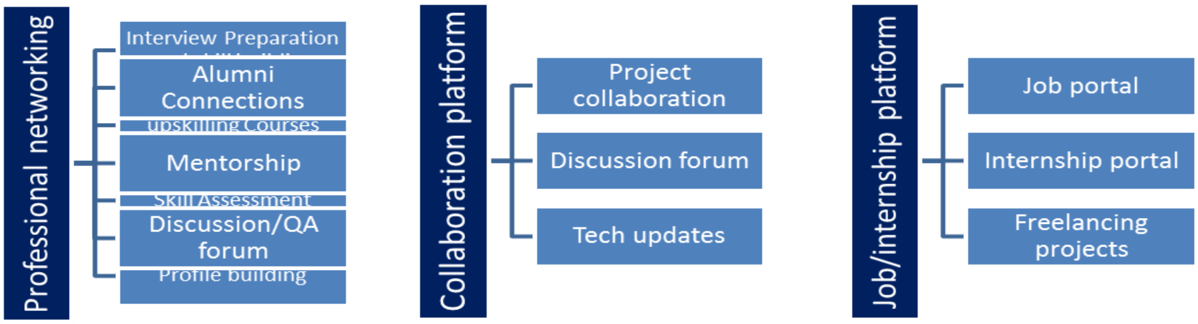
upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.

Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

https://www.upskillcampus.com/



Professional networking

- Interview Preparation
- Alumni Connections
- upskilling Courses
- Mentorship
- Skill Assessment
- Discussion/QA forum
- Profile building

Collaboration platform

- Project collaboration
- Discussion forum
- Tech updates

Job/internship platform

- Job portal
- Internship portal
- Freelancing projects

2.3 The IoT Academy

The IoT academy is the EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

☛ get practical experience of working in the industry.

☛ to solve real world problems.

☛ to have improved job prospects.

☛ to have Improved understanding of our field and its applications.

☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1]

[2]

[3]

2.6 Glossary

Terms	Acronym

3 Problem Statement

In the assigned problem statement

As cities grow, traffic congestion becomes a major problem, leading to delays, pollution, and economic loss. Manual traffic management is often inefficient, especially during peak hours and holidays. City authorities need a system that can accurately predict traffic volume in advance to manage resources effectively and plan for future infrastructure.

The goal of this project is to develop a machine learning model that can forecast the number of vehicles at four key city junctions based on historical data, including the time of day, day of the week, and public holidays.

4 Existing and Proposed solution

Existing Solution: Currently, traffic management is mostly reactive, relying on manual observation by traffic police or fixed-time traffic signals. This system struggles to adapt to real-time traffic fluctuations, special events, or holidays, leading to inefficiencies.

Proposed Solution: My proposed solution is a predictive model built using Python and the Random Forest algorithm. The model analyzes historical data to learn complex patterns related to time and junction characteristics.

Value Addition:

- **Proactive Management:** The model can predict traffic peaks hours or days in advance, allowing authorities to be prepared.
- **Data-Driven Decisions:** It provides a data-backed approach to traffic management, replacing guesswork.
- **Web Application:** I have also developed and deployed a user-friendly web application using Streamlit where anyone can input a date, time, and junction to get an instant traffic prediction. (Live App URL: https://trafficforcasting.streamlit.app/)

4.1 Code submission (Github link): Akpandey04/upskillcampus

4.2 Report submission (Github link) : upskillcampus/SmartCity_AdarshKumarPandey_USC_UCT.pdf at main · Akpandey04/upskillcampus

5 Proposed Design/ Model

The solution follows a standard machine learning workflow, which can be broken down into the following stages:

- **Data Loading and Preparation:** The historical traffic data (`train.csv` and `test.csv`) is loaded using the Pandas library. The 'DateTime' column is converted into the correct format for analysis.
- **Exploratory Data Analysis (EDA):** Before building the model, I analyzed the data to understand its patterns. This involved creating visualizations like:
- A box plot to compare traffic volume across different junctions.
- A bar plot to see the impact of holidays on traffic.
- This step confirmed that time-based features would be crucial for the model.
- **Feature Engineering:** New, informative features were created from the 'DateTime' column to help the model learn better. These include:
- Time-based features: `hour`, `dayofweek`, `month`, `year`.
- Event-based features: `is_holiday` and `is_weekend`.
- **Model Training:** The prepared data was used to train a `RandomForestRegressor` model. This model is powerful and works well with tabular data. The data was split chronologically to ensure the model was validated on unseen future data.
- **Model Evaluation:** The model's performance was tested on the validation set to ensure its accuracy and reliability.
- **Deployment:** The final, trained model was saved as a `.pkl` file and deployed as an interactive web application using Streamlit.

5.1 High Level Diagram (if applicable)

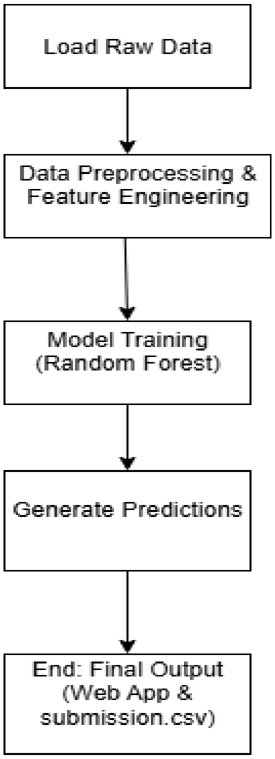

Load Raw Data
Data Preprocessing & Feature Engineering
Model Training (Random Forest)
Generate Predictions
End: Final Output (Web App & submission.csv)

Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

5.2 Low Level Diagram (if applicable)

5.3 Interfaces (if applicable)

Update with Block Diagrams, Data flow, protocols, FLOW Charts, State Machines, Memory Buffer Management.

6 Performance Test

The final model was evaluated using two key metrics on the validation set, which the model had never seen before:

- **R-squared (R^2): 0.9197**
  - This is an excellent score. It means that our model was able to successfully explain **91.97%** of the variations in the traffic data. This proves that the features we created were very effective and the model learned the patterns correctly.
- **Root Mean Squared Error (RMSE): 7.76**
  - This score means that, on average, the model's prediction for the number of vehicles is off by approximately **8 vehicles**. Given that traffic can range from very low to over 100 vehicles, an average error of just 8 is a very strong result.

These results confirm that the model is highly accurate and reliable for real-world use.

6.1 Test Plan/ Test Cases: The primary test plan was to evaluate the model's performance on a dataset that it has never seen before. To simulate a real-world forecasting scenario, the historical data was split chronologically, ensuring the model was tested on its ability to predict future traffic rather than just interpolating past data.

- **Test Case 1: Weekday Rush Hour:** Predicting traffic for a regular weekday (e.g., Monday, 9 AM). The expected outcome is a high traffic volume.
- **Test Case 2: Public Holiday:** Predicting traffic for a known public holiday (e.g., 15th August). The expected outcome is a traffic pattern different from a regular weekday.
- **Test Case 3: Weekend Night:** Predicting traffic for a weekend late at night (e.g., Sunday, 2 AM). The expected outcome is a very low traffic volume.

6.2 Test Procedure

- The historical data was split into a training set (data before April 2017) and a validation set (data from April 2017 onwards).
- The Random Forest model was trained exclusively on the training set.
- The trained model was then used to make predictions on the entire validation set.
- The model's predictions were compared against the actual known vehicle counts from the validation set.

- Performance was measured using R-squared (R²) and Root Mean Squared Error (RMSE) metrics to determine the model's accuracy and average error.

6.3 Performance Outcome: The final model was evaluated using two key metrics on the validation set, which the model had never seen before:

- R-squared (R²): 0.9197
  - This is an excellent score. It means that our model was able to successfully explain 91.97% of the variations in the traffic data. This proves that the features we created were very effective and the model learned the patterns correctly.
- Root Mean Squared Error (RMSE): 7.76
  - This score means that, on average, the model's prediction for the number of vehicles is off by approximately 8 vehicles. Given that traffic can range from very low to over 100 vehicles, an average error of just 8 is a very strong result.

These results confirm that the model is highly accurate and reliable for real-world use.

7 My learnings

This internship was an incredible learning experience. My key takeaways are:

- **Technical Skills:** I gained hands-on experience with Python and essential data science libraries like Pandas, NumPy, Scikit-learn, and Matplotlib/Seaborn.
- **Machine Learning Workflow:** I learned the complete end-to-end process of a real-world machine learning project, from data cleaning and feature engineering to model training, evaluation, and finally, deployment.
- **Time-Series Analysis:** I understood the importance of handling time-series data correctly, especially the concept of chronological validation instead of random splitting.
- **Model Deployment:** I learned how to save a trained model and deploy it as an interactive web application using Streamlit, making the model accessible to non-technical users.
- **Problem Solving:** This project taught me how to break down a complex problem into smaller, manageable steps and find effective solutions.
- In addition to the main Smart City Traffic Forecasting project, I took the initiative to further develop my machine learning skills by building two other foundational models. These projects allowed me to gain a broader perspective on different types of ML tasks and strengthen my core competencies.

1. House Price Predictor (Regression Task)

- **Objective:** To build a model that could accurately predict house prices based on various features like area, number of bedrooms, location, and amenities.
- **Skills Gained:** This project solidified my understanding of regression algorithms, feature scaling, and how to evaluate model performance using metrics like Mean Squared Error (MSE).
- **GitHub Link:** Akpandey04/House-Price-Prediction

2. Financial Fraud Detection (Classification Task)

- **Objective:** To develop a model capable of identifying fraudulent financial transactions from a large dataset of legitimate and fraudulent activities.
- **Skills Gained:** This project was crucial for learning about classification problems, especially how to handle imbalanced datasets (where fraud cases are very rare). I gained experience with techniques like SMOTE and evaluation metrics such as Precision, Recall, and F1-Score.
- **GitHub Link:** Akpandey04/Financial-Fraud-Detection

8 Future work scope

During this project, I realized there are several exciting ways this work could be taken to the next level. Given more time, I would explore the following ideas:

- **Using More Complex Models:** The Random Forest model worked very well, but for time-series data, deep learning models like LSTMs or GRUs are very powerful. It would be interesting to see if these models could capture more subtle, long-term trends in the traffic data that the current model might miss.
- **Adding Weather and Event Data:** A major factor that our current model doesn't account for is the real world outside of dates and times. Integrating an external weather API to add data like rainfall or temperature could significantly improve predictions, as bad weather often leads to more traffic. Similarly, adding a calendar of local events like concerts or cricket matches would help the model predict unusual traffic spikes.
- **Developing a Real-Time System:** The current Streamlit app is great for manual predictions, but a more advanced version could be connected directly to live traffic

camera feeds or road sensors. This would create a real-time dashboard for traffic authorities, allowing them to see live predictions and react instantly to changing conditions.