AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY
(AN AUTONOMOUS INSTITUTION)

Industrial Internship Report on

"Forecasting of Smart city traffic patterns"

Prepared by

Paladugu Vishnu Vardhan

Executive Summary
This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT). This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time. My project was (Tell about ur Project) This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

1 Preface

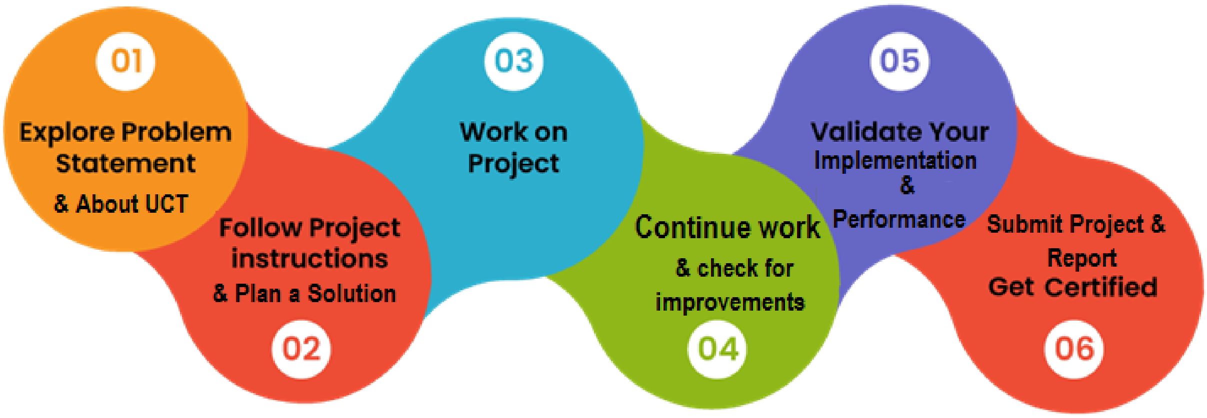
Summary of the whole 6 weeks' work.

About need of relevant Internship in career development.

Brief about Your project/problem statement.

Opportunity given by USC/UCT.

How Program was planned



Your Learnings and overall experience.

Thank to all (with names), who have helped you directly or indirectly.

Your message to your juniors and peers.

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.

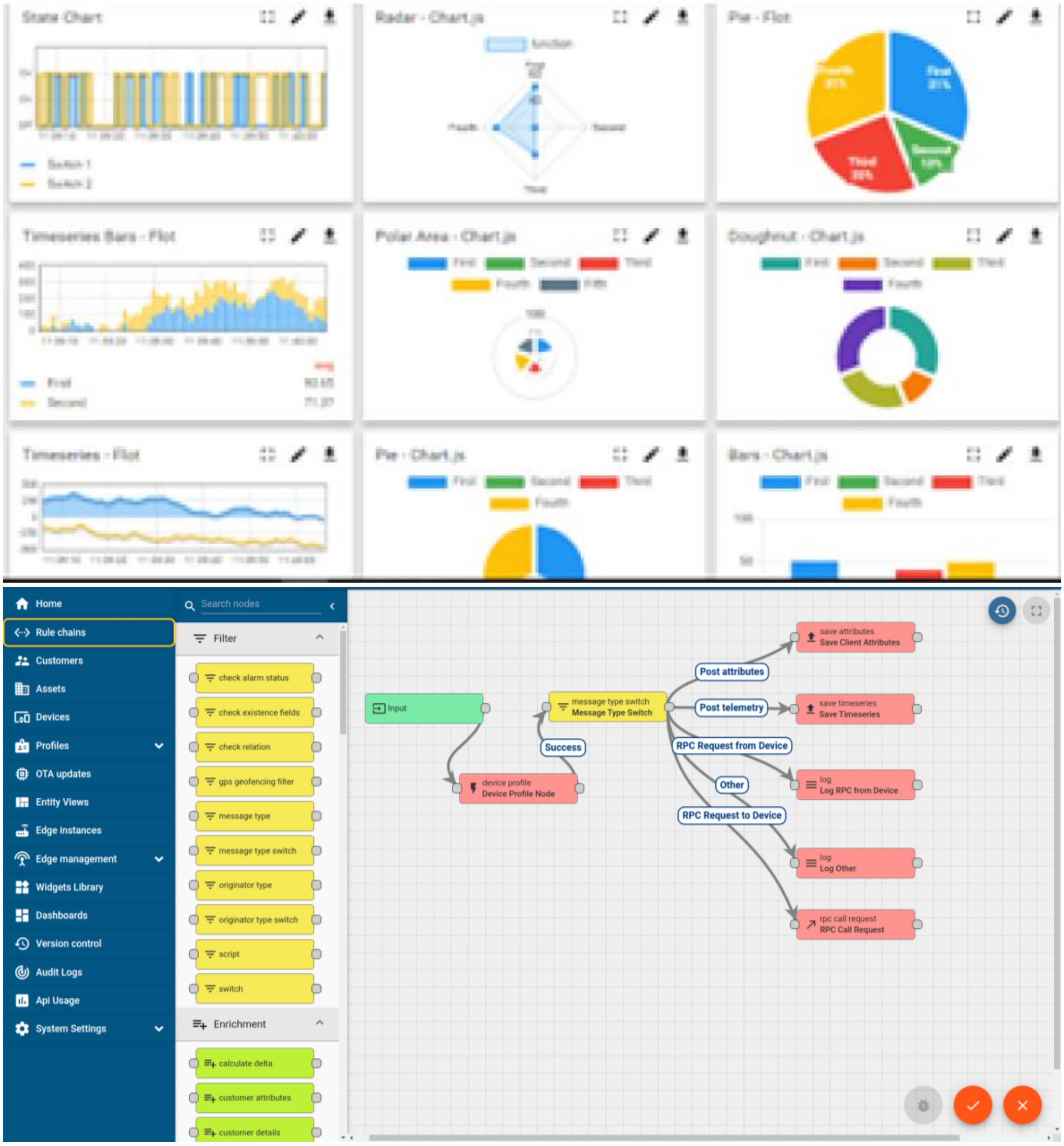
i. UCT IoT Platform (uct Insight)

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



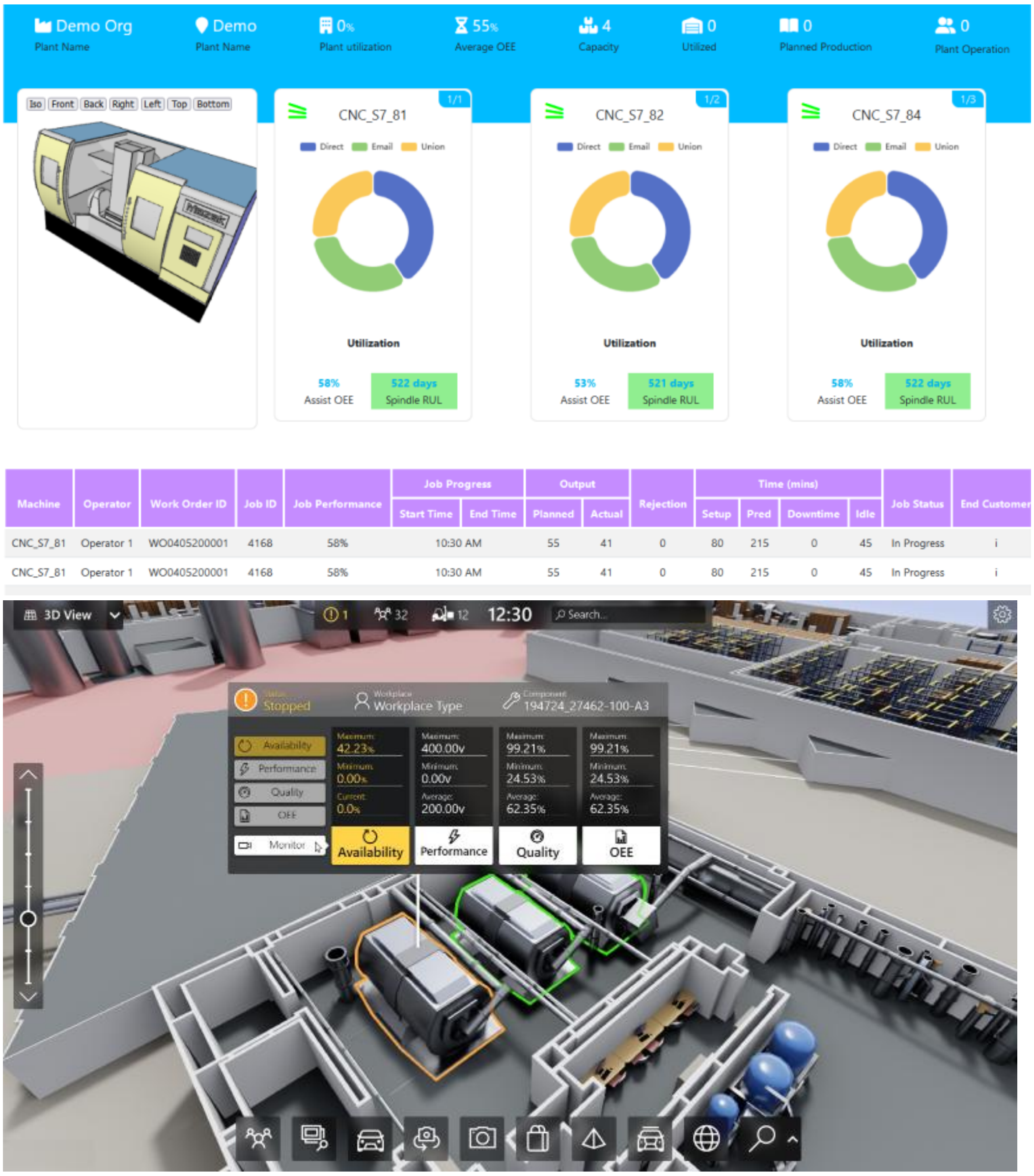
ii. Smart Factory Platform (FACTORY WATCH)

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 unleased 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.

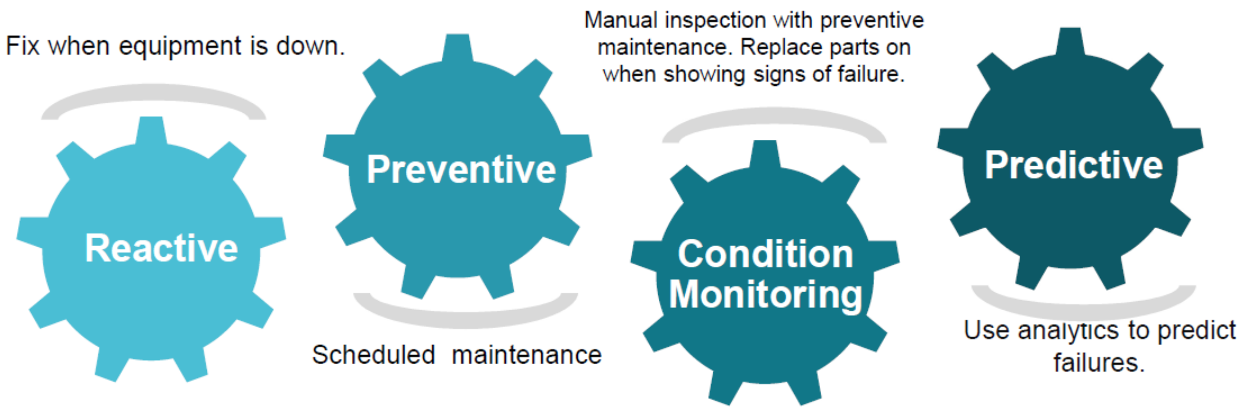


iii. LoRaWAN based Solution

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

iv. Predictive Maintenance

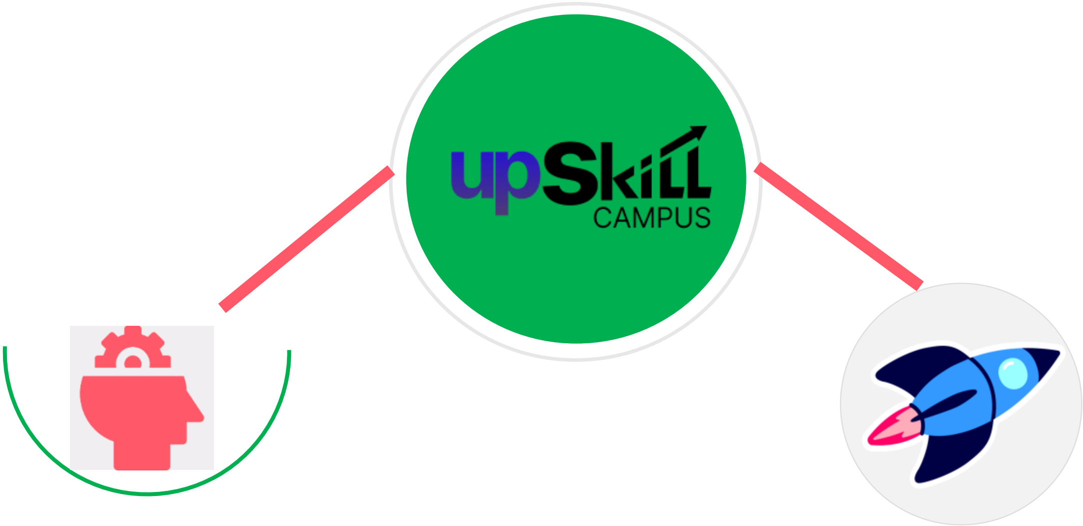
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

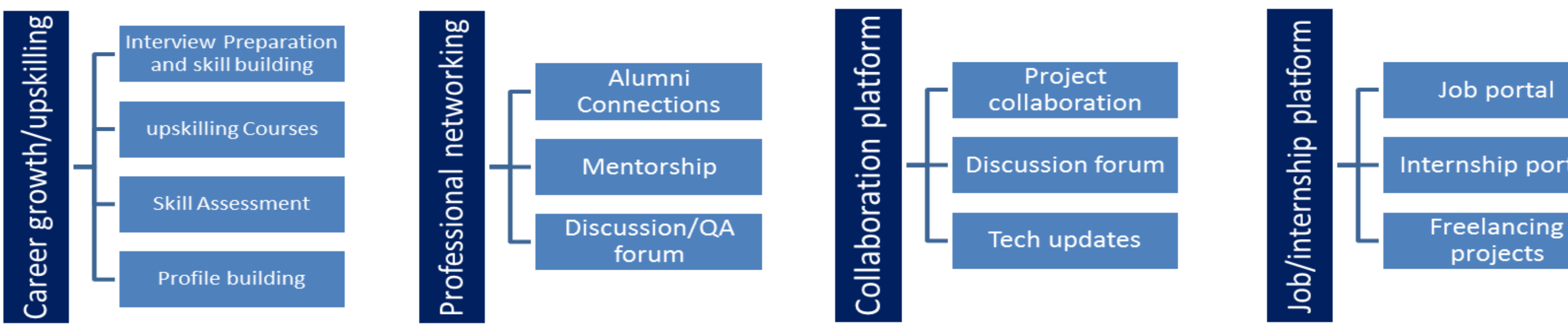
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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/



2.3 The IoT Academy

The IoT academy is 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] https://learn.upskillcampus.com/s/mycourses

[2] https://www.uniconvergetech.in/

[3] https://upskillcourses.com/

3 Problem Statement

With cities all around the world expanding and changing, traffic congestion is becoming a major problem. The growing number of cars on the road, urbanization, and the quickening pace of population growth all contribute to the worsening of this issue. Smart city projects are being put into action to use technology and data-driven solutions to solve these issues. The capacity to precisely estimate and predict traffic patterns is a critical component of smart city design. Therefore, the problem statement revolves around developing effective forecasting models for smart city traffic patterns. The objective is to develop a system that can predict traffic patterns in a smart city setting with accuracy. Predicting traffic congestion levels, journey durations, and traffic flow along different routes, junctions, and road networks are all part of this process.

4 Existing and Proposed solution

Summary of existing solutions and their limitations

1. Traditional Statistical Models
Existing approaches often rely on statistical techniques like autoregressive integrated moving average (ARIMA) or exponential smoothing.

- **Limitations:** They assume linear relationships and cannot capture the complex, nonlinear influences on traffic. They also struggle with large datasets and real-time updates, which restricts both their accuracy and scalability.

2. Machine Learning Approaches
Some systems forecast traffic using ML methods such as gradient boosting, random forests, and support vector machines (SVM).

- **Limitations:** These methods depend heavily on manual feature engineering—an often time-consuming process that requires domain expertise. They may also fail to fully model the dynamic, nonlinear interactions present in real traffic data, reducing prediction quality.

Proposed Solution:

This work applies advanced techniques to model traffic at four junctions, incorporating all relevant factors in the dataset. After evaluating several algorithms, I found that a Decision Tree provides the best forecasts for smart-city traffic patterns.

A Decision Tree model represents decisions in a tree structure:

- Internal nodes correspond to features
- Branches represent decision rules
- Leaves indicate outcomes (predicted traffic levels)

Algorithm steps:

1. **Data preparation:** Construct a labeled dataset linking input features (time, weather, etc.) to traffic outcomes.
2. **Attribute selection:** Choose splits that maximize label purity in each subset.
3. **Recursive partitioning:** Divide data by the chosen feature at each node until a stopping criterion is met.
4. **Missing-value handling:** Incorporate strategies to manage incomplete records.
5. **Prediction:** Traverse the tree for new inputs to forecast traffic.

The decision tree algorithm offers a flexible and understandable method for predicting traffic patterns, allowing you to obtain knowledge of the underlying variables affecting traffic and anticipate future behavior using data from the past.

4.1 Code submission (Github link):

https://github.com/Ver-er/upskillcampus/blob/main/Forecasting_of_smart_city_traffic_patterns.ipynb

4.2 Report submission (Github link):

https://github.com/Ver-er/upskillcampus/blob/main/Forecasting_of_smart_city_traffic_patterns_Vishnu_USC_UCT.pdf

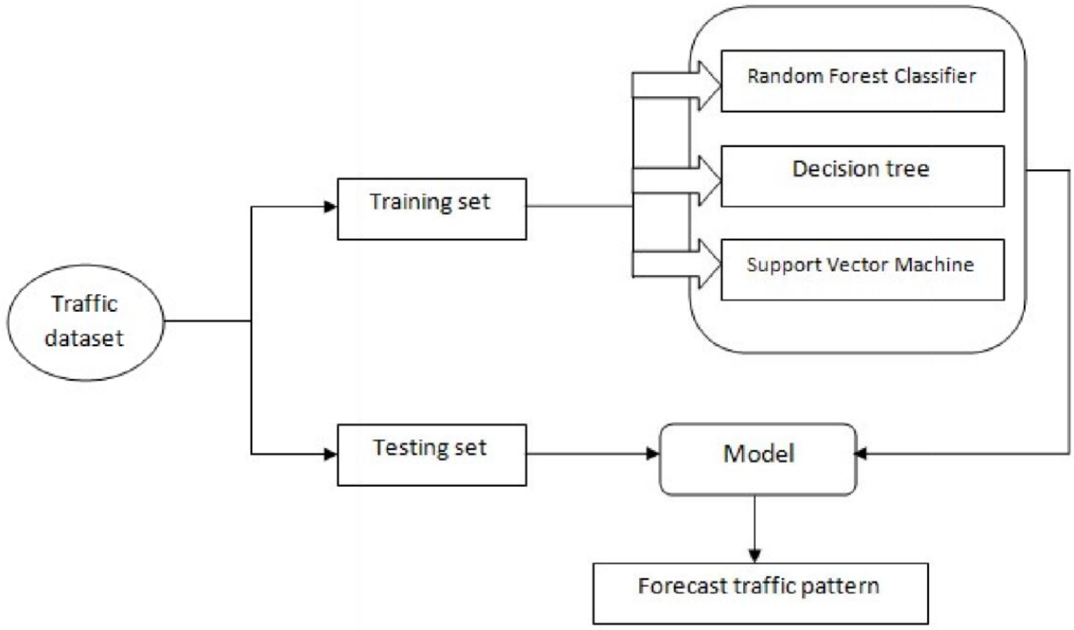
5 Proposed Design/ Model

A step-by-step workflow for ML-based traffic forecasting:

1. **Data Collection:** Acquire historical traffic records with features such as time of day, day of week, weather, holidays, construction, etc.
2. **Data Preprocessing:** Cleanse data (remove duplicates, impute missing values, address outliers) and engineer informative features.
3. **Train/Test Split:** Partition data into training and testing sets.
4. **Model Training:** Build the Decision Tree (or alternative algorithms), selecting splits by information-gain criteria; tune hyperparameters to prevent over- or underfitting.
5. **Evaluation:** Test on held-out data, computing metrics like mean absolute error (MAE) or accuracy.
6. **Deployment:** Retrain on the full dataset and deploy the model; schedule periodic updates as new data arrives.

7. **Prediction & Monitoring:** Continuously forecast traffic for new inputs and monitor model performance, adjusting as needed.

5.1 Interfaces



6 Performance Test

The Forecasting Smart City Traffic Patterns model cannot be successfully predicted without the use of predictive models. With the use of cutting-edge machine learning methods and resources, we set out to build a solid model that could examine the traffic patterns. To validate our forecasting model, we used scikit-learn's train_test_split to separate training and evaluation data. Random splitting ensured robust pattern learning while preserving unseen samples for testing. Preprocessing (one-hot encoding, normalization) was handled via scikit-learn transformers.

We assessed models using cross-validation and metrics such as mean squared error (MSE), mean absolute error (MAE), and R^2 score.

6.1 Test Plan/ Test Cases

Test Plan for Forecasting Traffic Patterns model:

1. Test Objective: Verify the precision and dependability of different machine learning algorithms for traffic pattern forecasting model.
2. Test Environment:
   - Programming language: Python
   - Libraries: scikit-learn, Pandas, NumPy
   - Traffic data set: Historical traffic data containing features such as date, time, weather conditions, etc.
3. Test Data:
   - Prepare a dataset with historical traffic data, including known patterns and corresponding outcomes.
   - Split the dataset into training and testing sets (e.g., 70% training, 30% testing).
4. Test Cases:

   a. Data Preprocessing:

   i. Verify that the dataset is loaded correctly, and all required features are present. ii. Check for missing or invalid values and ensure proper handling or imputation. iii. Validate the normalization or scaling of numerical features if applicable.

   b. Model Training:

   i. Train the model using the training dataset.
   ii. Verify that the model has been trained successfully without any errors.
   iii. Validate that the model has learned patterns and relationships from the training data.

   c. Model Evaluation:

   i. Apply the trained model to the testing dataset.

ii. Compare the predicted traffic patterns with the actual patterns in the testing dataset. iii. Calculate evaluation metrics such as accuracy, precision, recall, and F1-score. iv. Ensure that the model performance meets the defined acceptance criteria.

d. Model Validation:

i. Use cross-validation techniques (e.g., k-fold cross-validation) to assess the model's generalization capabilities. ii. Verify that the model performs consistently across different subsets of the data.

iii. Ensure that the model does not overfit or underfit the training data.

e. Performance Testing:

i. Measure the training and prediction times to ensure they are within acceptable limits.

ii. Evaluate the model's performance on large datasets to validate scalability.

f. Boundary and Edge Cases:

i. Test the model's behavior with extreme or outlier values in the input features. ii. Verify that the model handles unexpected or novel traffic patterns gracefully.

g. Integration Testing:

i. Validate the integration of the Decision Tree algorithm with other components or systems.

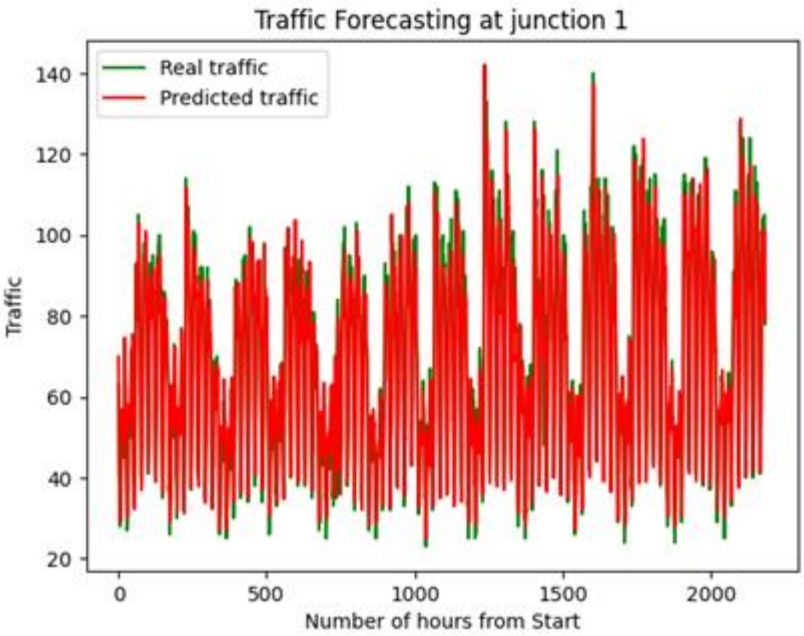
ii. Verify the compatibility and data exchange between the traffic forecasting model and external systems.

6.2 Performance Outcome

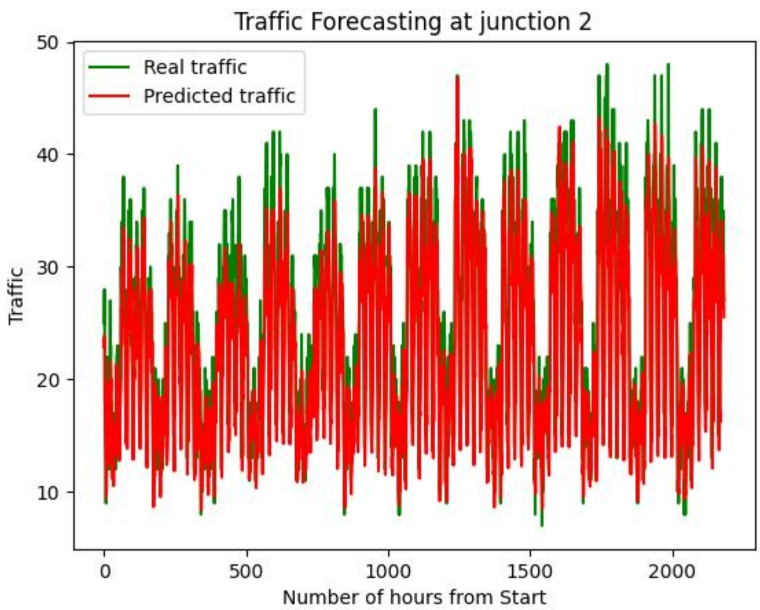
The performance outcome of my forecasting traffic patterns model can vary depending on several factors, including the quality of the data, the complexity of the traffic patterns, the choice of features, and the tuning of the model parameters.

Visualizing Prediction of traffic at different Junctions:

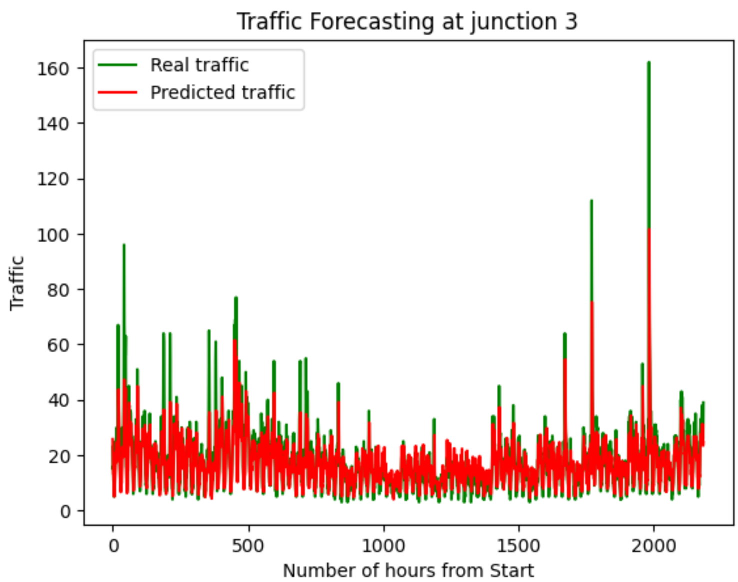
- Junction 1:



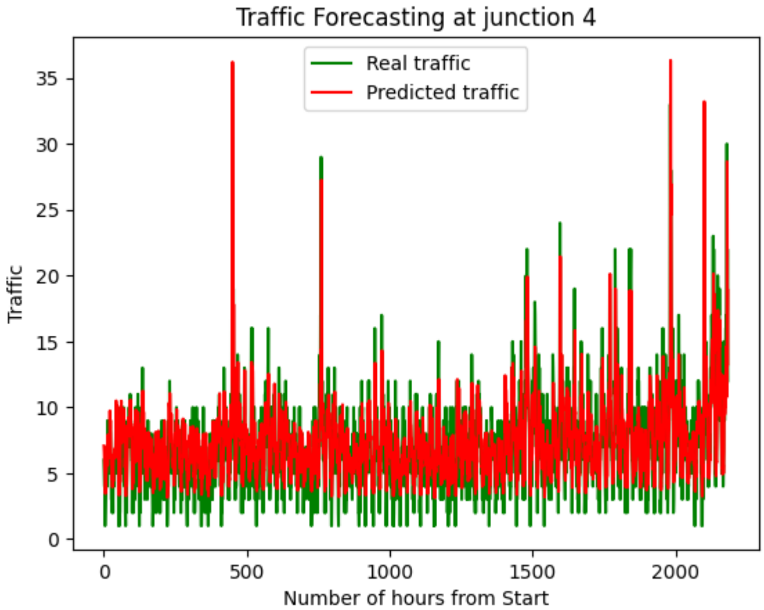
- Junction 2:



- Junction 3:



- Junction 4:



Accuracy of different algorithms:

- Random forest classifier:

  *Accuracy score for RF: 79.46799667497922

- Decision tree classifier:

  *Accuracy score for DT:
  79.46799667497922

- Support Vector Machine (SVM):

  *Accuracy score for SVM: 95.36159600997507

7 My learnings

During my six-week internship in data science and machine learning, I had the opportunity to work on a project focused on Forecasting of Smart city traffic patterns. This project not only exposed me to the intricate idea of Smart city but also allowed me to apply various data science techniques and machine learning algorithms to solve real-world problems. By applying data science techniques and machine learning algorithms to real-world problems, I not only enhanced my technical skills but also gained a deeper appreciation for the role of data-driven solutions in addressing societal challenges. This internship has inspired me to continue exploring the intersection of data science.

8 Future work scope

The Support Vector Machine has shown promising results in improving algorithmic efficiency and accuracy. However, there is room for improvement in fine-tuning hyperparameters and exploring ensemble techniques. Integrating real-time data sources, such as live GPS feeds, weather conditions, and special events, can enhance the predictive capabilities of the models. AI-based anomaly detection can help identify unusual traffic patterns or incidents, contributing to proactive traffic management and emergency response systems. Expanding the model to multiple cities and integrating it with smart traffic signal systems can lead to automated adjustments in signal timings, reducing congestion and optimizing traffic flow.

Limitations include data quality and availability, static feature set, model complexity, external factors, and human behaviour variability. Accurate traffic prediction relies on highquality and diverse datasets, but the model's limitations highlight the need for ongoing development to create a comprehensive and adaptable traffic management solution

As we look towards the future, the potential for enhancing predictive models for forecasting traffic patterns is vast. Building upon the foundation laid by our existing model development efforts, several avenues for further research and innovation emerge. Additionally, the integration of a user interface (UI) adds a layer of accessibility and usability.