**Industrial Internship Report on**

**”Forecasting Of Smart City Traffc Pattern”**

**Prepared by**

**Princy**

|  |
| --- |
| *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 about forecasting smart city traffic patterns in which we are working with the government to transform our city into a smart city. The vision is to convert it into a digital and intelligent city to improve the efficiency of services for the citizens. One of the problems faced by the government is traffic. You are a data scientist working to manage the traffic of the city better and to provide input on infrastructure planning for the future.  The government wants to implement a robust traffic system for the city by being prepared for traffic peaks. They want to understand the traffic patterns of the four junctions of the city.  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 3](#_Toc139702806)

[2 Introduction 5](#_Toc139702807)

[2.1 About UniConverge Technologies Pvt Ltd 5](#_Toc139702808)

[2.2 About upskill Campus 9](#_Toc139702809)

[2.3 Objective 11](#_Toc139702810)

[2.4 Reference 11](#_Toc139702811)

[2.5 Glossary 11](#_Toc139702812)

[3 Problem Statement 12](#_Toc139702813)

[4 Existing and Proposed solution 13](#_Toc139702814)

[5 Proposed Design/ Model 14](#_Toc139702815)

[6 Performance Test 15](#_Toc139702819)

[6.1 Test Plan/ Test Cases 16](#_Toc139702820)

[6.2 Test Procedure 18](#_Toc139702821)

[6.3 Performance Outcome 19](#_Toc139702822)

[7 My learnings 20](#_Toc139702823)

[8 Future work scope 21](#_Toc139702824)

# Preface

## Traffic affects every citizen’s life in many ways by how long it takes for him or her to travel from home to office, the air condition he or she inhales, the strain generated by traffic jams, sleep, and workouts induced by time spent in traffic. Since motorists cannot see the entire traffic system, the urban traffic system must be anticipated in order to sensitize residents about their mobility choices and the subsequent impact on the environment, as well as to implement smart transport system.

Among the most difficult problems affecting cities is designing and developing an effective transportation system. As the population of the city grows, so do the city’s governmental and non-governmental transport networks, and even minor faults and accidents can have a negative impact on a system that is already stretched to its limits. Traffic jam and delays on transportation systems can cost a city losses in terms of economy and have a negative impact on inhabitants’ standard of living as they spend a lot of time traveling from one location to another. Traffic gridlock has a number of negative consequences, including lost time, environmental degradation, and safety hazards, as well as a negative influence on economic growth and a deterioration of individuals’ relationship with their municipal authorities. It is required that cities address fundamental concerns such as the amount of time drivers spend looking for parking spaces, enhancing highway safety, guaranteeing that public transportation is properly routed, and allowing citizens to travel by many modes of transportation. The idea smart cities became a reality thanks to advancement in Information Technology (IT). Various types of transport models are routinely employed in the estimation of traffic status.

First, a comparison is made using different simple regression techniques to determine the best performing model. Second, synchronous bagging and stacking techniques were used to find a more accurate model of the two comparisons. The results showed that the K-Nearest Neighbors (KNN) bagging group performed significantly better than all other regression models used in this study. Experimental results show that the KNN bagging ensemble model helps to reduce the error rate by more than 30% in predicting traffic congestion

Data science and machine learning work together to give valuable insights in some real life scenarios. Doing an internship in DS and ML helps to properly understand and find solutions to real life problems by implementing it.

How Program was planned



Thanks to all guiding hands at upskill, who have helped me directly or indirectly.

# Introduction

## 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.



1. UCT IoT Platform **(****)**

**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

 

1. **Smart Factory Platform (****)**

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.

 

1.  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.

1. 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.



## 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

[https://www.upskillcampus.com/](about:blank)

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



## 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.

## 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.

## Reference

[1] <https://www.theiotacademy.co/>

[2] https://upskillcourses.com/

## Glossary

|  |  |
| --- | --- |
| Terms | Acronym |
| ensemble | a number of things considered as a group a set of clothes that are worn together |
|  |  |
|  |  |
|  |  |
|  |  |

# Problem Statement

You are working with the government to transform your city into a smart city. The vision is to convert it into a digital and intelligent city to improve the efficiency of services for the citizens. One of the problems faced by the government is traffic. You are a data scientist working to manage the traffic of the city better and to provide input on infrastructure planning for the future.

The government wants to implement a robust traffic system for the city by being prepared for traffic peaks. They want to understand the traffic patterns of the four junctions of the city. Traffic patterns on holidays, as well as on various other occasions during the year, differ from normal working days.

This is important to take into account for your forecasting.

The problem statement revolves around developing an effective forecasting traffic patterns for reducing problems such as congestion levels , travelling time and traffic etc. the accuracy and reliability of the forecasting models can be evaluated using methods such as mean absolute error(MAE), root mean square error(RMSE), or mean absolute percentage error(MAPE). These quantify the differences between the predicted and actual traffic patterns.

# Existing and Proposed solution

The proposed Smart city traffic pattern was borne out of the problem associated with urban city. Smart cities use big data for decision-making and a revolution in the Internet of a Thing (IoT) allows such data to be collection and transmission of such data. Transportation is a fundamental element influencing metropolitan areas and a crucial use case for smart cities.

The predicting modeling is the ware house of machine learning such as KNN, BAG used in this research. The accuracy of the machine learning is evaluated in the performance evaluation section.

Existing solutions for forecasting traffic patterns vary in their approaches and limitations. The challenges may include real time data updates , dynamic data, data quality etc.

I have decided to us Decision tree algorithm for forecasting traffic patterns. It helps in data preparation, handling missing values and handling numerical and categorical data, pruning, prediction, ensemble techniques , feature importance etc.

The decision tree algorithm offers a flexible and understandable method fr forecasting traffic patterns, allowing us to obtain knowledge of the underlying variables affecting traffic patterns and predict future behavior using historical or past data.

## Code submission (Github link)

## Report submission (Github link) :

# Proposed Design/ Model

Proposed design flow:

1. Data Collection:

Gather historical traffic data.

1. Data Preprocessing:

Clean the data by removing duplicates, handling missing values, and addressing outliers. Perform feature engineering to extract meaningful features. Convert categorical variables into numerical representations.

1. Splitting the Data:

Divide the preprocessed data into training and testing sets.

4.Training the Decision Tree Model:

Apply the decision tree algorithm to the training data.Select the best feature to split the data at each internal node based on criteria like information gain.

5.Model Evaluation:

Evaluate the trained decision tree model using the testing data.Calculate metrics such as accuracy, precision, or mean absolute error (MAE).

6.Model Deployment:

Deploy the decision tree model for traffic pattern forecasting using the entire dataset for training.

7. Prediction and Monitoring:

Use the deployed decision tree model to forecast traffic patterns based on new input data.

# Performance Test

The decision tree algorithm's architecture of a traffic forecasting system can be significantly impacted by identified constraints. Following are some typical restrictions and suggestions about how to handle them:

1. Limited Data Availability: The quality and dependability of the forecasting model may be hampered by a lack of past traffic data. To manage this constraint, some suggestions are as follows:

* Acquiring additional data: See if there is a way to collect more historical traffic data, either by expanding the time frame or by obtaining information from several sources.
* Data augmentation: Consider data augmentation methods like simulation or synthetic data generation to expand the dataset's size and diversity if getting new data is not practical.
* Feature engineering: To obtain the most information possible from the little data, concentrate on extracting pertinent aspects.

1. Incomplete or Noisy Data: The decision tree model's performance may be significantly impacted by poor data quality, missing values, or outliers. To manage this constraint, some suggestions are as follows:

* Data cleaning: To deal with missing values, outliers, and inconsistencies in the dataset, use effective data cleaning procedures. Think about outlier identification algorithms to handle extreme values and imputation techniques for handling missing numbers.
* Feature selection: To reduce the impact of noisy or irrelevant data, give preference to the selection of robust and relevant attributes.
* Robust modeling techniques: Utilise noise-resistant decision tree variations, such random forests or gradient boosting, which can manage noisy data and reduce overfitting.

1. Computation Time and Resource Constraints: The decision tree approach can be computationally and resource-intensive, especially when working with huge datasets or complicated features. To manage this constraint, some suggestions are as follows:

* Sample or subset the data: Consider sampling or sub setting strategies to lessen processing costs while maintaining data integrity if the dataset is too vast.
* Feature reduction: Reduce the number of features and make the decision tree construction process simpler by using feature selection or dimensionality reduction approaches.
* Model optimization: To establish a balance between accuracy and computing economy, optimize the decision tree algorithm's hyperparameters and look for the optimum configurations.
* Distributed computing or parallelization: Use parallelization methods or distributed computing frameworks to quicken the training and assessment procedures, especially when working with huge datasets.

1. Interpretability Requirements: The model's interpretability may be a constraint in some circumstances. Although interpretability of decision trees is fundamental, complex trees with numerous layers might be challenging to comprehend. To manage this constraint, some suggestions are as follows:

* Limiting the tree depth: Control the decision tree's maximum depth to produce a model that is easier to understand. Pruning methods or hyperparameter manipulation can be used to achieve this.
* Ensembling with simpler models: To keep the decision tree model understandable while increasing accuracy, combine it with other straightforward, comprehensible models like rule-based models or linear regression.

To make wise design choices, it is crucial to recognise and comprehend the unique restrictions and requirements of the traffic forecasting problem.

## Test Plan/ Test Cases

1. Test Objective: Verify the precision and dependability of the Decision Tree algorithm-based traffic pattern forecasting model.
2. Test Environment:

* Programming language: Python
* Libraries: scikit-learn, pandas, numpy, matplotlib
* Traffic data set: Historical traffic data containing features such as date, time, weather conditions, etc.

1. 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).

1. 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 Decision Tree 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.

## Test Procedure

1. Set up the Test Environment

* Install the required software and libraries (Python, scikit-learn, pandas, numpy, matplotlib).
* Set up the development environment with the necessary dependencies.
* Ensure that the historical traffic dataset is available for testing.

1. Identify Test Data:

* Select a subset of the historical traffic dataset for testing.
* Split the dataset into training and testing sets (e.g., 70% training, 30% testing).

1. Preprocessing:

* Load the training dataset into the system.
* Perform any necessary preprocessing steps, such as handling missing values, normalizing or scaling features, and encoding categorical variables.

1. Model Training:

* Implement the Decision Tree algorithm using the scikit-learn library.
* Train the Decision Tree model using the training dataset.
* Ensure that the training process completes without any errors or exceptions.
* Validate that the model has been trained successfully by inspecting its attributes and structure.

1. Model Evaluation:

* Load the testing dataset into the system.
* Apply the trained Decision Tree model to the testing dataset to predict traffic patterns.
* Compare the predicted traffic patterns with the actual patterns in the testing dataset.
* Calculate evaluation metrics such as accuracy, precision, recall, and F1-score to assess the model's performance.
* Ensure that the model meets the defined acceptance criteria.

1. Performance Testing:

* Measure the training and prediction times of the model to ensure they meet the performance requirements.
* Evaluate the model's performance on larger datasets to assess scalability.

## Performance Outcome

The accuracy of forecasting traffic patterns is an extremely important measure of performance. It can be evaluated by comparing the predicted traffic patterns to the patterns that are actually observed. A higher level of accuracy signifies a superior performance outcome.

The performance outcome of a Decision Tree algorithm for forecasting traffic patterns 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 to avoid overfitting and underfitting.

# My learnings

We learned about the python programming and its various implementation in areas like Artificial Intelligence, Data Science and Machine Learning and also about different algorithms. We get to know the importance of Machine learning in today’s world. We learned how to apply different algorithms and how its implementation can be done, and which will give the best result on implementation. We get to used pandas library in python to read our data and perform data cleaning, then we use scikit-learn library of machine learning to split train and test data and then apply the machine learning algorithm and we use matplotlib and seaborn for graphical representation of data to analyse and understand it.

This 6 Week Internship program with UpSkill Campus &UniConverge Technologies Pvt. Ltd. was very helpful. I learned a lot about Pyhton, Machine learning, Data Science and working on jupyter and get to work on a real world project. It was a great experience which will help me get ahead in my career in future. Thanks to UpSkil Campus and UCT for giving us this opportunity.

**8 Future work scope**

Incorporating real-time data:

Decision trees can be trained using historical traffic data, but real-time data, such as traffic flow, weather, and events, may increase predicting accuracy. The decision tree may adjust and create predictions based on the present circumstances by incorporating these variables, producing more precise traffic forecasts.

Dynamic updating:

Various variables, like urban development, modifications to the road system, or significant events, can cause changes in traffic patterns throughout time. Consider adding a dynamic updating mechanism to the decision tree model to take these changes into account. Retrain the decision tree on a regular basis with the most recent data to account for changing traffic patterns and guarantee forecast accuracy.