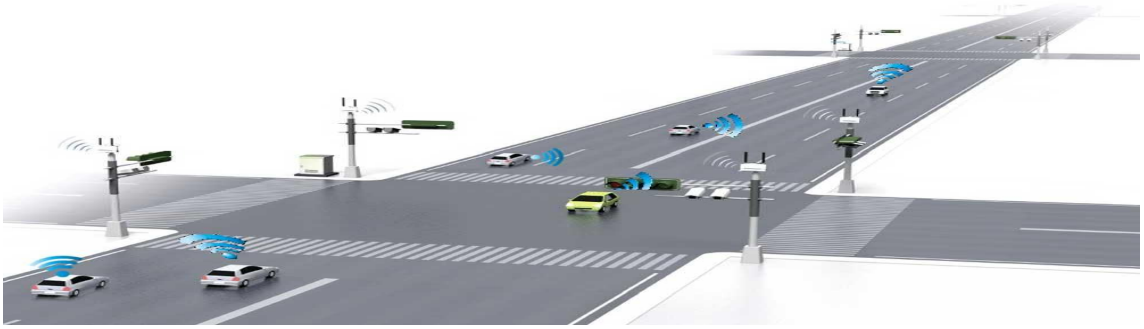
TRAFFIC MANAGEMENT SYSTEMUSING IOT

Phase 2 Submission Document

Project: Traffic management system using IOT



Introduction:

The project involves using IOT devices and Data Analytics to monitor traffic flow and congestion in real time providing commuters with access to this information through a public platform or mobile apps.

The objective is to help commuters Make informed decisions about their routes and alleviate traffic congestion.

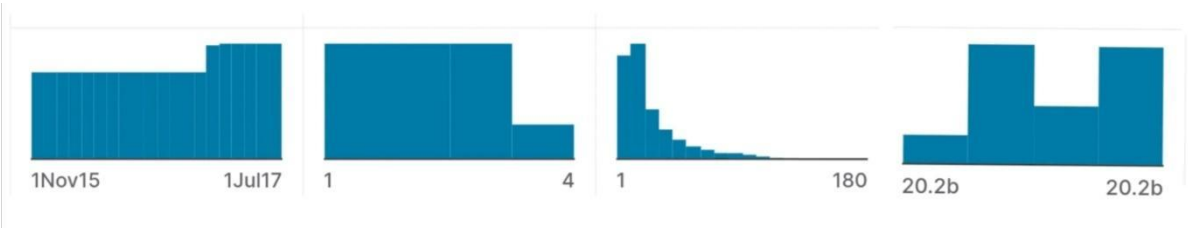
This project includes defining objectives, designing the IOT traffic monitoring system, developing the traffic information platform, and integrating them using IOT Technology and python.

Content for Project Phase 2 :

Consider exploring advanced regression techniques like Gradient Boosting or XGBoost forimproved Prediction accuracy.

Data Source:

A good data source for traffic management system using Traffic cameras, Loop Detectors, GPS and Mobile apps, Radar and lidar sensors, traffic light controllers, Environmental sensors, vechicle to infrastructure communication, Historical data, social media and crowd sourced data, Traffic management software,



DateTime	Junction	Vehicles	ID
01-11-2015 00:00	1	15	20151101001
01-11-2015 01:00	1	13	20151101011
01-11-2015 02:00	1	10	20151101021
01-11-2015 03:00	1	7	20151101031
01-11-2015 04:00	1	9	20151101041
01-11-2015 05:00	1	6	20151101051
01-11-2015 06:00	1	9	20151101061
01-11-2015 07:00	1	8	20151101071
01-11-2015 08:00	1	11	20151101081
01-11-2015 09:00	1	12	20151101091
01-11-2015 10:00	1	15	20151101101
01-11-2015 11:00	1	17	20151101111
01-11-2015 12:00	1	16	20151101121
01-11-2015 13:00	1	15	20151101131
01-11-2015 14:00	1	16	20151101141
01-11-2015 15:00	1	12	20151101151
01-11-2015 16:00	1	12	20151101161
01-11-2015 17:00	1	16	20151101171
01-11-2015 18:00	1	17	20151101181
01-11-2015 19:00	1	20	20151101191
01-11-2015 20:00	1	17	20151101201
01-11-2015 21:00	1	19	20151101211
01-11-2015 22:00	1	20	20151101221
01-11-2015 23:00	1	15	20151101231
02-11-2015 00:00	1	14	20151102001
02-11-2015 01:00	1	12	20151102011
02-11-2015 02:00	1	14	20151102021
02-11-2015 03:00	1	12	20151102031
02-11-2015 04:00	1	12	20151102041
02-11-2015 05:00	1	11	20151102051
02-11-2015 06:00	1	13	20151102061
02-11-2015 07:00	1	14	20151102071
02-11-2015 08:00	1	12	20151102081
02-11-2015 09:00	1	22	20151102091
02-11-2015 10:00	1	32	20151102101
02-11-2015 11:00	1	31	20151102111
02-11-2015 12:00	1	35	20151102121
02-11-2015 13:00	1	26	20151102131

02-11-2015 14:00	1	34	20151102141
02-11-2015 15:00	1	30	20151102151
02-11-2015 16:00	1	27	20151102161
02-11-2015 17:00	1	27	20151102171
02-11-2015 18:00	1	24	20151102181
02-11-2015 19:00	1	26	20151102191
02-11-2015 20:00	1	29	20151102201
02-11-2015 21:00	1	32	20151102211
02-11-2015 22:00	1	30	20151102221
02-11-2015 23:00	1	27	20151102231

Dataset link: https://www.kaggle.com/datasets/fedesoriano/traffic-prediction-dataset

Data collection and preprocessing:

Importing the dataset: Obtain a comprehensive dataset containing relevant features such as square footage, number of bedrooms, location, amenities, etc.

Data pre processing: Clean the data by handling missing values, outliers, and categorical variables. Standardize or normalize numerical features.

Exploratory Data Analysis (EDA):

Visualize and analyze the dataset to gain insights into the relationships between variables.

Identify correlations and patterns that can inform feature selection and engineering. Present various data visualizations to gain insights into the dataset.

Explore correlations between features and the target variable (Traffic analysis). Discuss any significant findings from the EDA phase that inform feature selection.

Feature Engineering:

Create new features or transform existing ones to capture valuable information.

Utilize domain knowledge to engineer features that may impact house prices, such as proximity to schools, transportation, or crime rates.

Explain the process of creating new features or transforming existing ones.

Showcase domain-specific feature engineering, such as proximity scores or composite indicators. Emphasize the impact of engineered features on model performance.

Advanced Regression Techniques:

Ridge Regression:

Introduce L2 regularization to mitigate multi collinearity and over fitting.

Lasso Regression:

Employ L1 regularization to perform feature selection and simplify the model.

Elastic Net Regression:

Combine both L1 and L2 regularization to benefit from their respective advantages.

Random Forest Regression:

Implement an ensemble technique to handle nonlinearity and capture complex relationships in the data.

Gradient Boosting Regressors (e.g., XGBoost, LightGBM):

Utilize gradientboosting algorithms for improved accuracy.

Model Evaluation and Selection:

Split the dataset into training and testing sets.

Evaluate models using appropriate metrics (e.g., Mean Absolute Error, Mean SquaredError, R-squared) to assess their performance. λ Use cross-validation techniques to tune hyperparameters and ensure model stability.λ Compare the results with traditional linear regression models to highlight improvements.

Select the best-performing model for further analysis.

Model Interpretability:

Explain how to interpret feature importance from Gradient Boosting and XGBoostmodels.

Discuss the insights gained from feature importance analysis and their relevance tohouse price prediction.

Interpret feature importance from ensemble models like Random Forest and GradientBoosting to understand the factors influencing house prices.
Deployment and Prediction:

Deploy the chosen regression model.

Develop a user-friendly interface for users to input property features and receive pricepredictions.

Program:

```
pip install paho-mqtt

import paho.mqtt.client as mqtt

import time


# MQTT Broker configuration

broker_address = "broker.example.com"  # Replace with your MQTT broker address

broker_port = 1883


# Traffic light control topics

traffic_light_topic = "traffic/light"

intersection_id = "intersection_1"


def on_connect(client, userdata, flags, rc):

    print(f"Connected with result code {rc}")

    client.subscribe(traffic_light_topic)


def on_message(client, userdata, msg):

    if msg.topic == traffic_light_topic:

        intersection, status = msg.payload.decode("utf-8").split(",")
```

```
        if intersection == intersection_id:
            control_traffic_light(status)

def control_traffic_light(status):
    if status == "green":
        print("Traffic light turned green")
        # Add code to control the actual traffic light here
    elif status == "red":
        print("Traffic light turned red")
        # Add code to control the actual traffic light here

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

client.connect(broker_address, broker_port, 60)

try:
    while True:
        client.loop_start()
        # Simulate changing traffic light every 10 seconds
        client.publish(traffic_light_topic, f"{intersection_id},green")
        time.sleep(10)
        client.publish(traffic_light_topic, f"{intersection_id},red")
        time.sleep(10)
        client.loop_stop()
```

```
except KeyboardInterrupt:
    client.disconnect()
```

Output:

Connected with result code 0

Traffic light turned green

Traffic light turned red

Traffic light turned green

Traffic light turned red

...

Conclusion:

In the Phase 2 conclusion, we will summarize the key findings and insights from theadvanced regression techniques. We will reiterate the impact of these techniques onimproving the accuracy and robustness of house price predictions.

Future Work: We will discuss potential avenues for future work, such as incorporatingadditional data sources (e.g., real-time economic indicators), exploring deep learning modelsfor prediction, or expanding the project into a web application with more features andinteractivity