

FINAL PROJECT

AI1801 - DAP391m- Instructor: HoangTN

Group 3

Nguyễn Vũ Huy - Leader

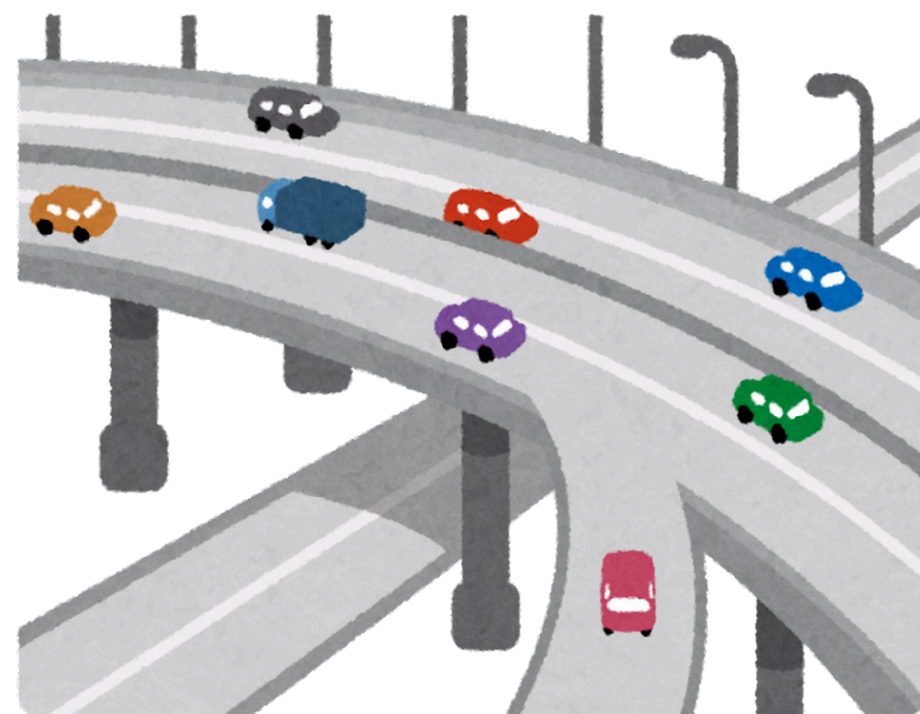
Nguyễn Thị Bích Tuyền

Nguyễn Khánh Trình

Đinh Văn Anh Khôi

Đặng Hoàng Kiệt

Nguyễn Phạm Thiên Phú



CONTENTS

- 1** **Introducing data**
- 2** **Data Visualization**
- 3** **Models used in the program**
- 4** **Deployment**
- 5** **Demo**

Introducing data

Traffic Dataset Overview:

Description: Collected by a computer vision model that detects four vehicle types (cars, bikes, buses, trucks). Stored in CSV format with data on time, date, day of the week, and vehicle count.

Key fields: CarCount, BikeCount, BusCount, TruckCount, Total (all vehicles every 15 minutes), and Traffic Status (1-Heavy, 2-High, 3-Normal, 4-Low).

1	Time,Date,Day of the week,CarCount,BikeCount,BusCount,TruckCount,Total,Traffic Situation
2	12:00:00 AM,10,Tuesday,31,0,4,4,39,low
3	12:15:00 AM,10,Tuesday,49,0,3,3,55,low
4	12:30:00 AM,10,Tuesday,46,0,3,6,55,low
5	12:45:00 AM,10,Tuesday,51,0,2,5,58,low
6	1:00:00 AM,10,Tuesday,57,6,15,16,94,normal
7	1:15:00 AM,10,Tuesday,44,0,5,4,53,low
8	1:30:00 AM,10,Tuesday,37,0,1,4,42,low
9	1:45:00 AM,10,Tuesday,42,4,4,5,55,low
10	2:00:00 AM,10,Tuesday,51,0,9,7,67,low
11	2:15:00 AM,10,Tuesday,34,0,4,7,45,low
12	2:30:00 AM,10,Tuesday,45,0,1,1,47,low
13	2:45:00 AM,10,Tuesday,45,0,1,3,49,low
14	3:00:00 AM,10,Tuesday,50,0,3,0,53,low
15	3:15:00 AM,10,Tuesday,34,0,4,4,42,low
16	3:30:00 AM,10,Tuesday,129,22,42,1,194,heavy
17	3:45:00 AM,10,Tuesday,144,16,49,0,209,heavy
18	4:00:00 AM,10,Tuesday,111,28,20,3,162,normal
19	4:15:00 AM,10,Tuesday,67,11,10,16,104,normal
20	4:30:00 AM,10,Tuesday,65,24,7,16,112,normal
21	4:45:00 AM,10,Tuesday,94,27,7,16,144,normal
22	5:00:00 AM,10,Tuesday,94,20,8,7,129,normal
23	5:15:00 AM,10,Tuesday,67,29,5,10,111,low
24	5:30:00 AM,10,Tuesday,56,12,2,14,84,normal

Introducing data

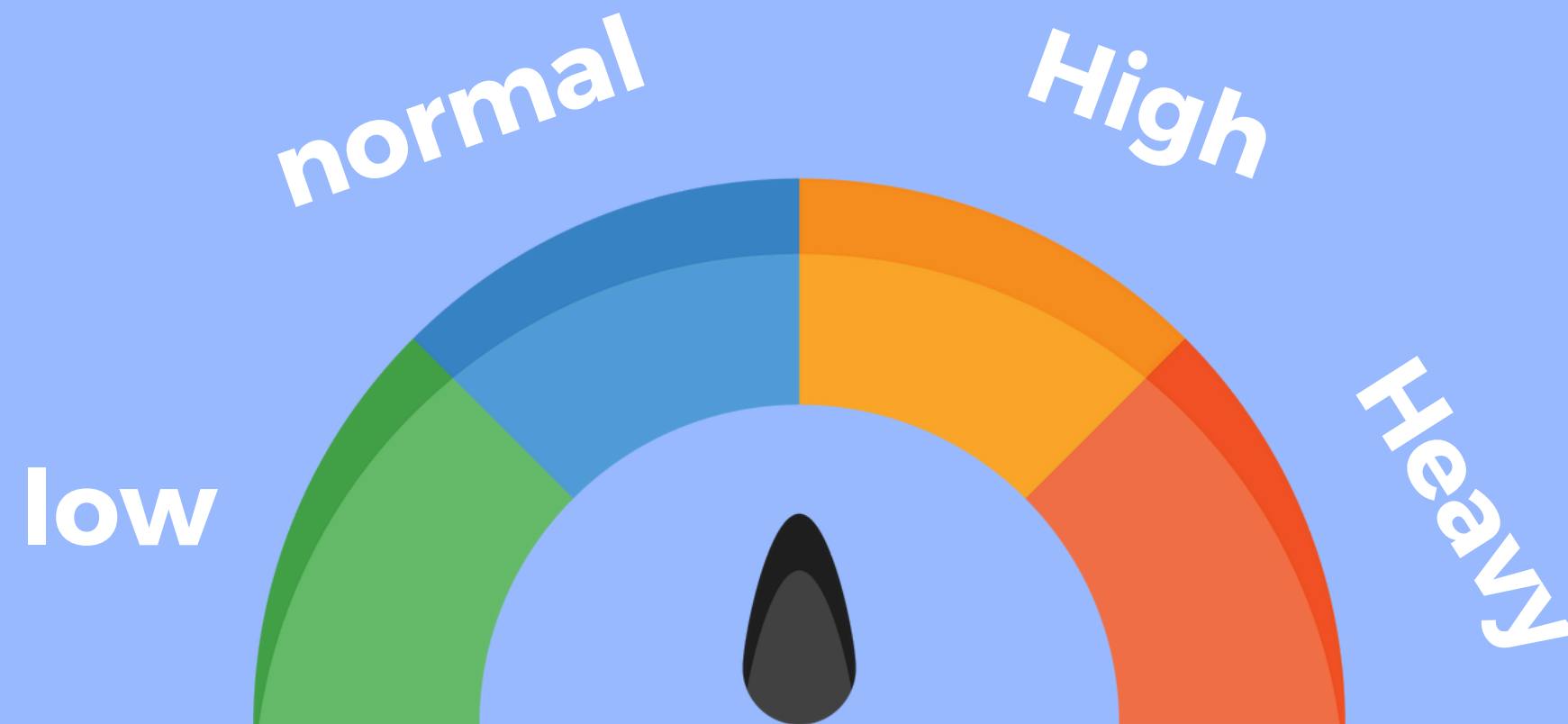
Use cases:

Traffic planning: Analyze traffic patterns, manage congestion, optimize signals, and adjust lanes.

Urban planning: Assess the impact of traffic on infrastructure, zoning, and congestion mitigation decisions.

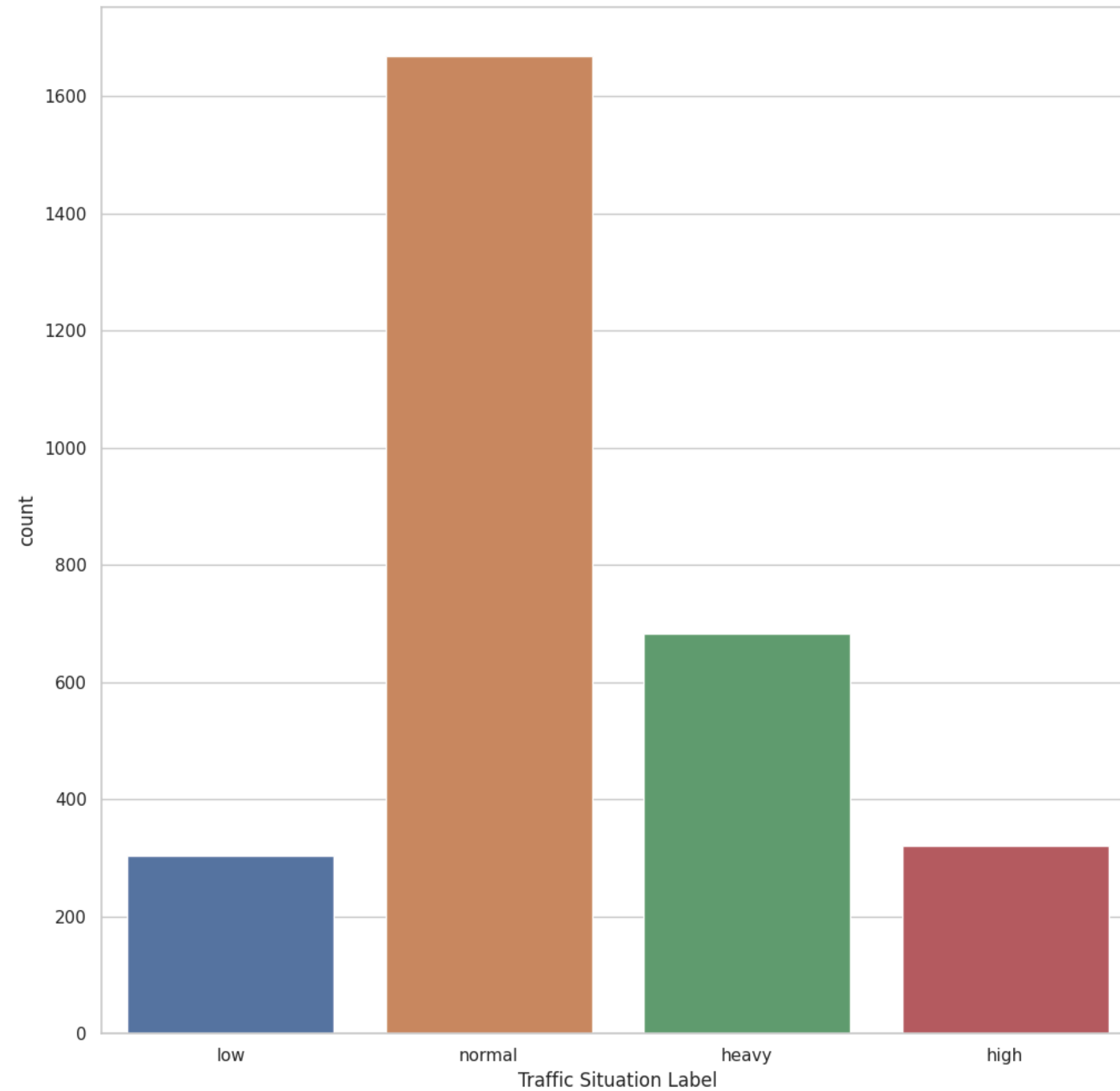
Traffic Study: Study hourly, daily, and daily patterns and explore correlations with external factors for better traffic flow insights.

=> in this project we use to predict traffic levels



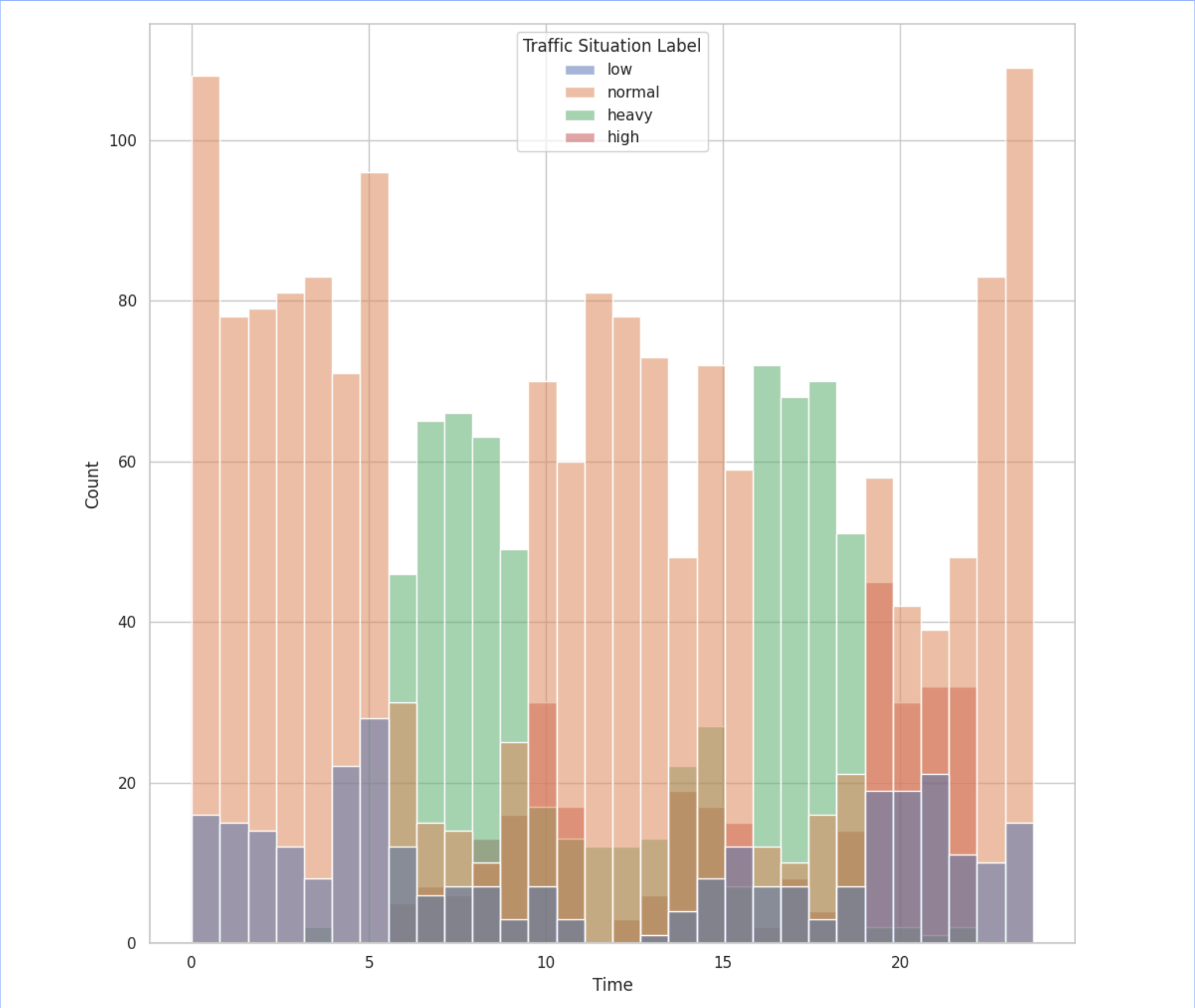
Data Visualization

traffic frequency



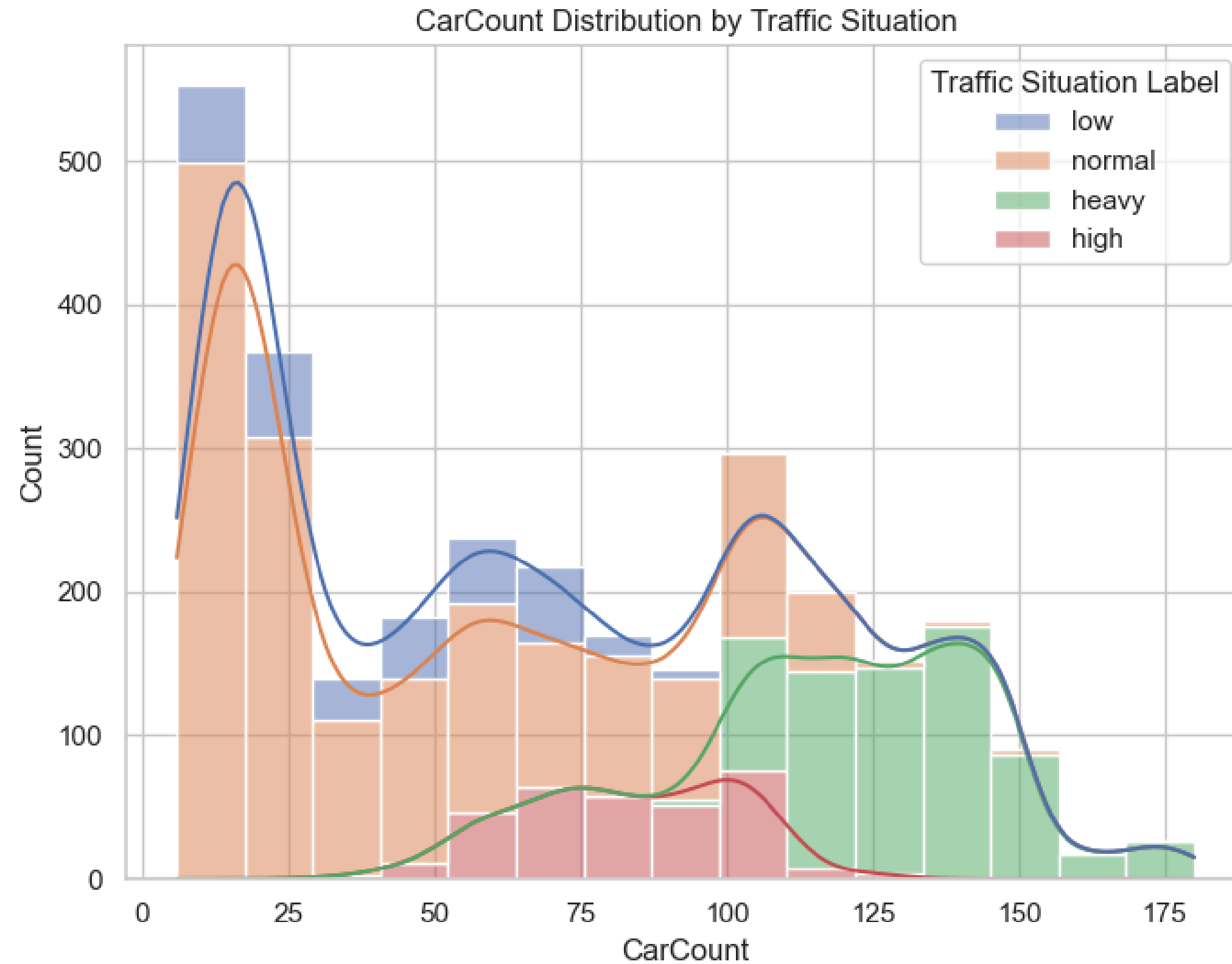
Data Visualization

traffic time and frequency



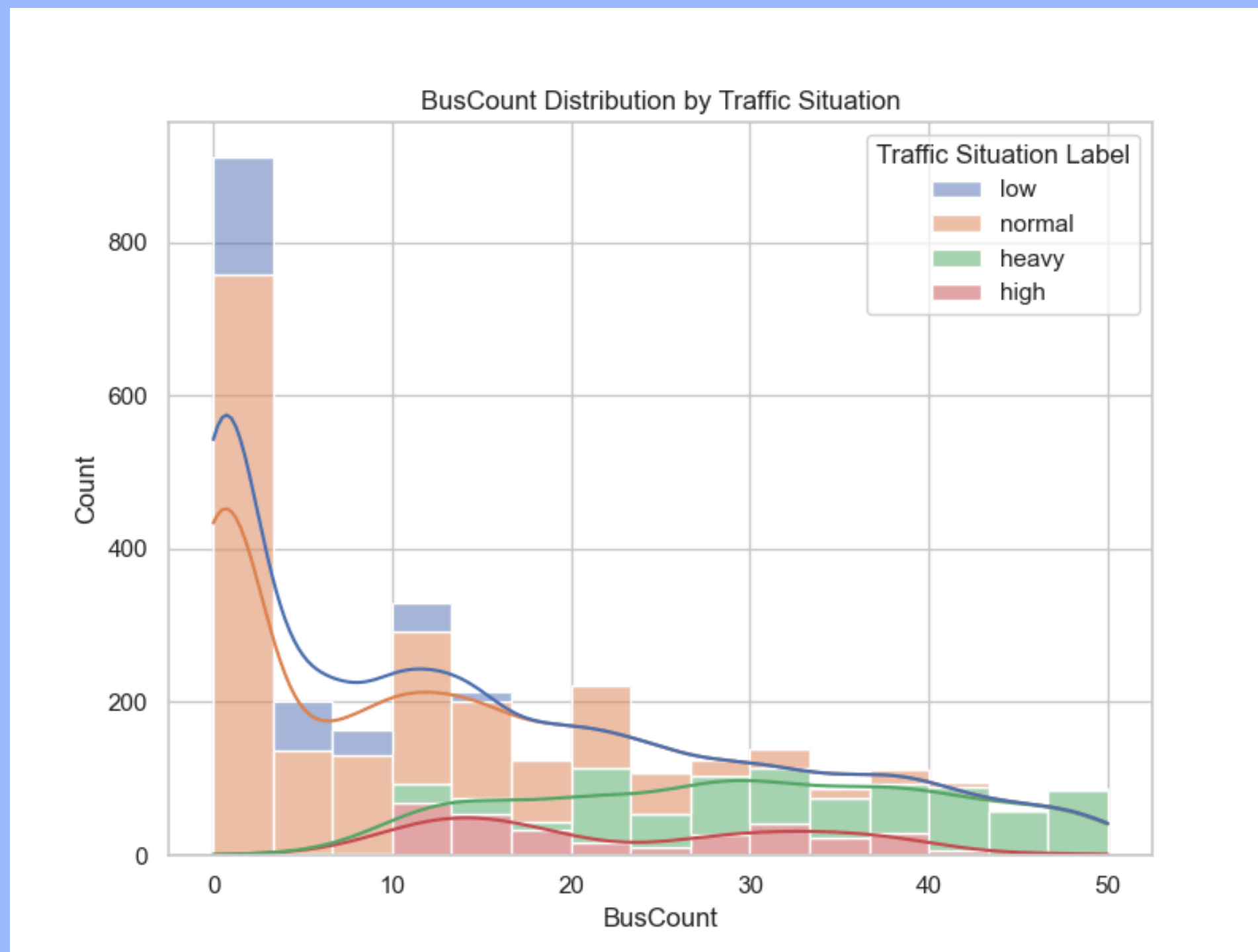
Data Visualization

vehicle traffic

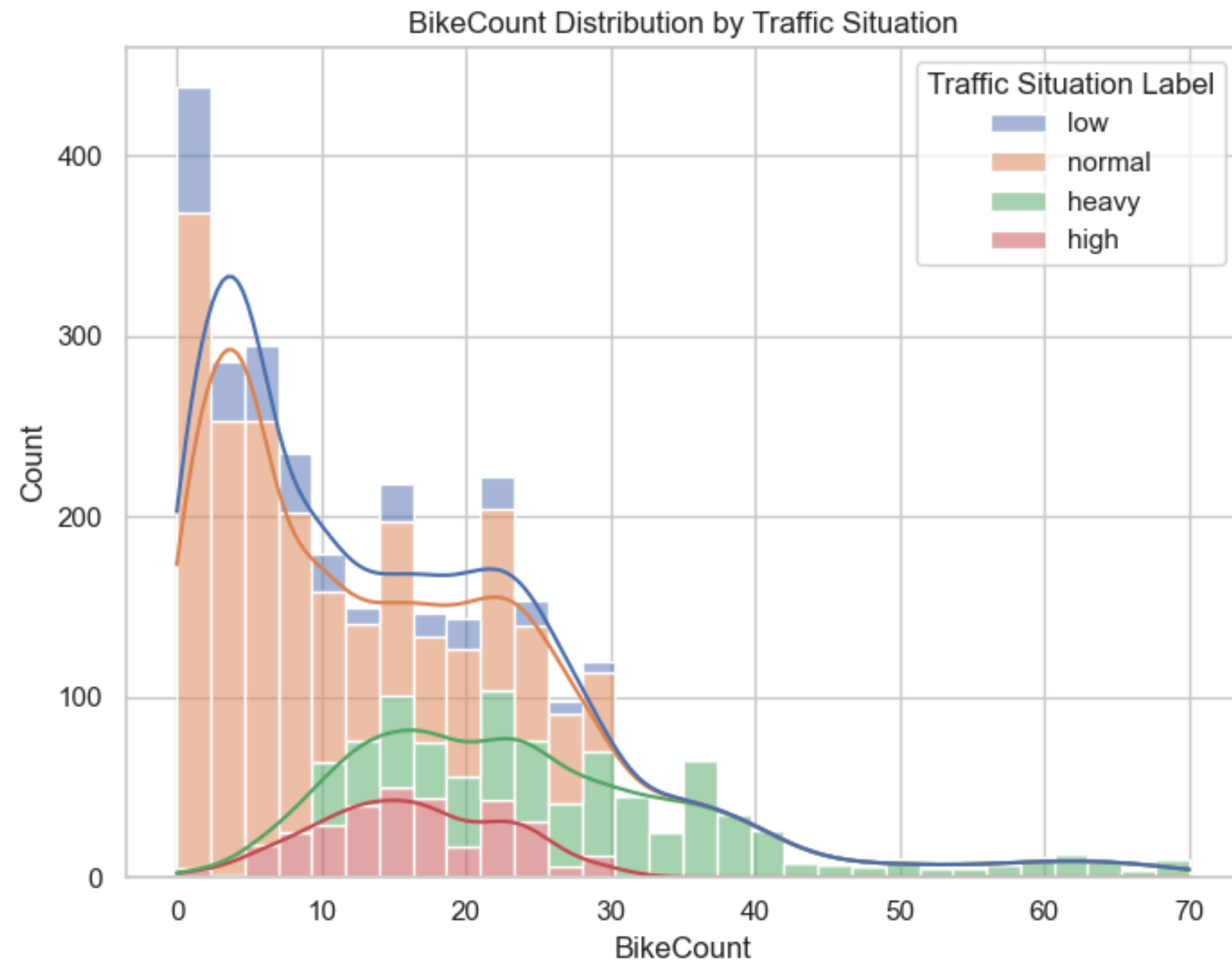


Data Visualization

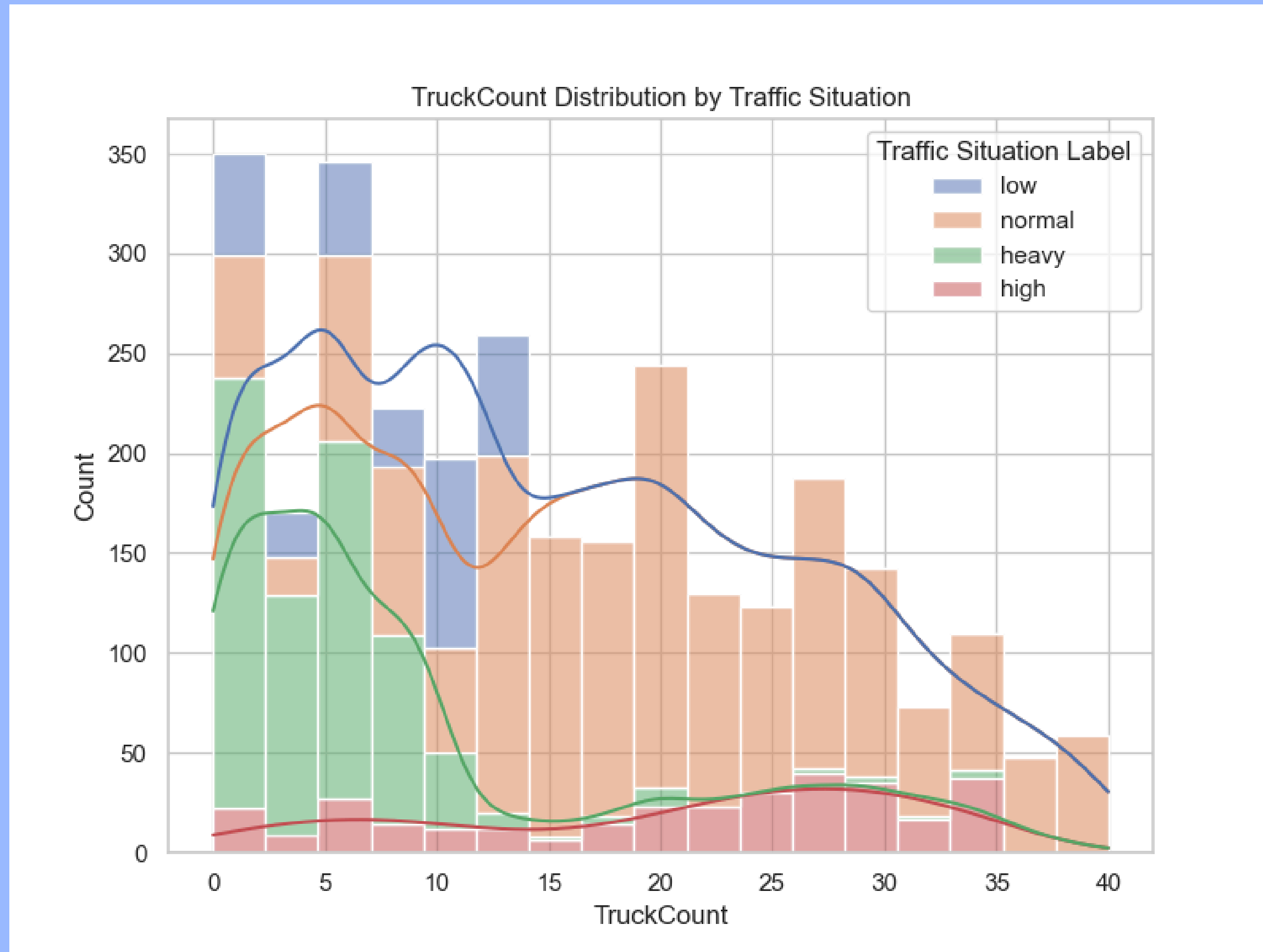
vehicle traffic



vehicle traffic



vehicle traffic



Models used in the program

Select Model

Random Forest

Random Forest

Gradient Boosting

Support Vector Machine

Knearest Neighbors

Logistic Regression

Naive Bayes

Neural Network

Car Count

Bike Count

Bus Count

Truck Count

Total Vehicle Count

Day of the Week

Monday

Predict

Select Model

Gradient Boosting

Date

Time (HH:MM:SS AM/PM)

e.g. 02:30:00 PM

Car Count

Bike Count

Bus Count

Truck Count

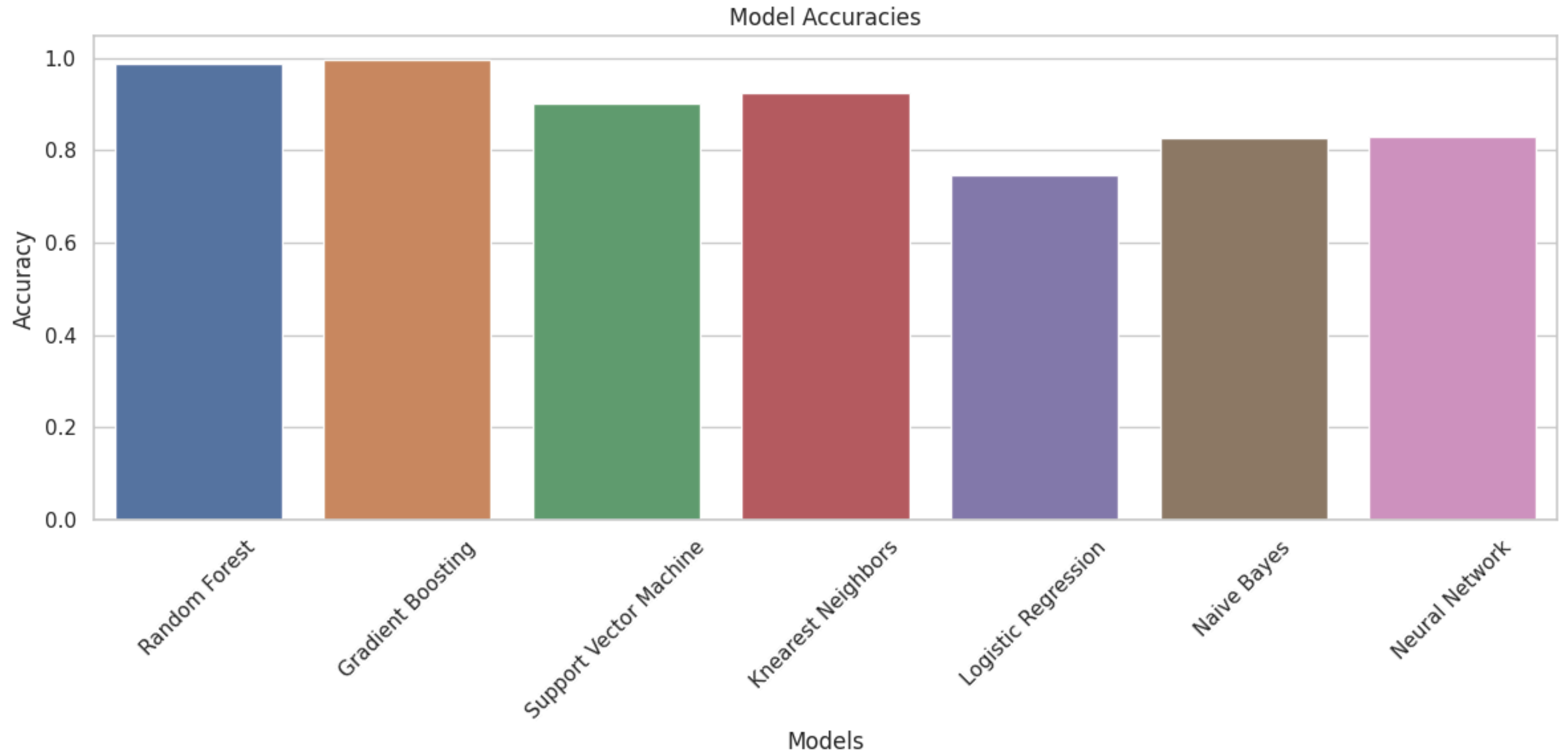
Total Vehicle Count

Day of the Week

Monday

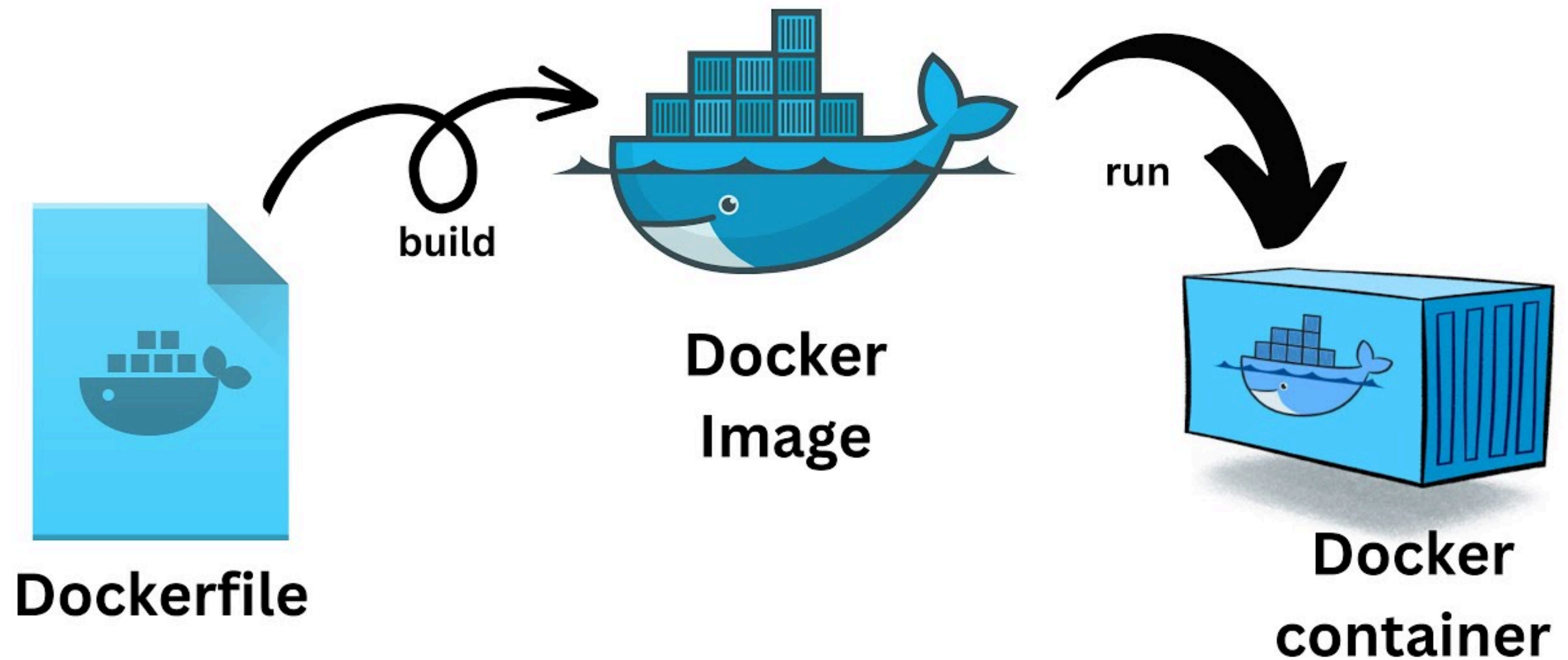
Predict

accuracy



Deployment with AWS - Docker

Docker file basics



ECR

Push commands for huy-ecr

macOS / Linux

Windows

Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see [Getting Started with Amazon ECR](#).

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see [Registry Authentication](#).

1. Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS CLI:

```
aws ecr get-login-password --region ap-southeast-1 | docker login --username AWS --password-stdin 354918392038.dkr.ecr.ap-southeast-1.amazonaws.com
```

Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.
2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](#). You can skip this step if your image is already built:

```
docker build -t huy-ecr .
```
3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag huy-ecr:latest 354918392038.dkr.ecr.ap-southeast-1.amazonaws.com/huy-ecr:latest
```
4. Run the following command to push this image to your newly created AWS repository:

```
docker push 354918392038.dkr.ecr.ap-southeast-1.amazonaws.com/huy-ecr:latest
```

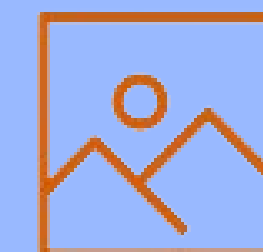
Close



Amazon Elastic Container Registry (Amazon ECR)



Registry



Image



ELASTIC
CONTAINER

ECR

ECR



FPT UNIVERSITY

[Alt+S]



Singapore ▾

vuhuyng ▾

Route 53 Elastic Beanstalk Lambda Amazon SageMaker CloudFront DynamoDB Elastic Container Service

Image scan overview, status, and full vulnerabilities has moved to the Image detail page. To access, click an image tag.

[Amazon ECR](#) > [Private registry](#) > [Repositories](#) > huy-ecr

huy-ecr

View push commands

Images (3)



Delete

Details

Scan

Search artifacts

< 1 >

<input type="checkbox"/>	Image tag ▾	Artifact type	Pushed at ▾	Size (MB) ▾	Image URI	Digest
<input type="checkbox"/>	latest	Image Index	November 06, 2024, 12:28:41 (UTC+07)	465.65	Copy URI	sha256:a6bfb5d0c93421384078ec3e03e07a...
<input type="checkbox"/>	-	Image	November 06, 2024, 12:28:41 (UTC+07)	0.00	Copy URI	sha256:e14ae842b004535fbe85155cdd6a8...
<input type="checkbox"/>	-	Image	November 06, 2024, 12:28:41 (UTC+07)	465.65	Copy URI	sha256:502b3fb03262adf3d4ab44f2b5f8aa...



ECS



FPT UNIVERSITY



Amazon Elastic Container Service (Amazon ECS)



Container 1



Container 2



Container 3



Task



Service



ECS Service Connect

aws

Services

Search

[Alt+S]

EC2

S3

IAM

RDS

Route 53

Elastic Beanstalk

Lambda

Amazon SageMaker

CloudFront

DynamoDB

Elastic Container Service

Amazon Elastic Container Service

Clusters

Namespaces

Task definitions

Account settings

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

Documentation

Discover products

Subscriptions

Amazon Elastic Container Service > Task definitions > DAP391m-Huy > Revision 1 > Containers

DAP391m-Huy:1

Deploy Actions Create new revision

Overview Info

ARN

arn:aws:ecs:ap-southeast-1:354918392038:task-definition/DAP391m-Huy:1

Task role

ecsrole

Status

ACTIVE

Task execution role

ecsTaskExecutionRole

Time created

November 06, 2024 at 12:30 (UTC+7:00)

Operating system/Architecture

Linux/X86_64

App environment

Fargate

Network mode

awsvpc

Containers

JSON

Task placement

Volumes (0)

Requires attributes

Tags

Task size

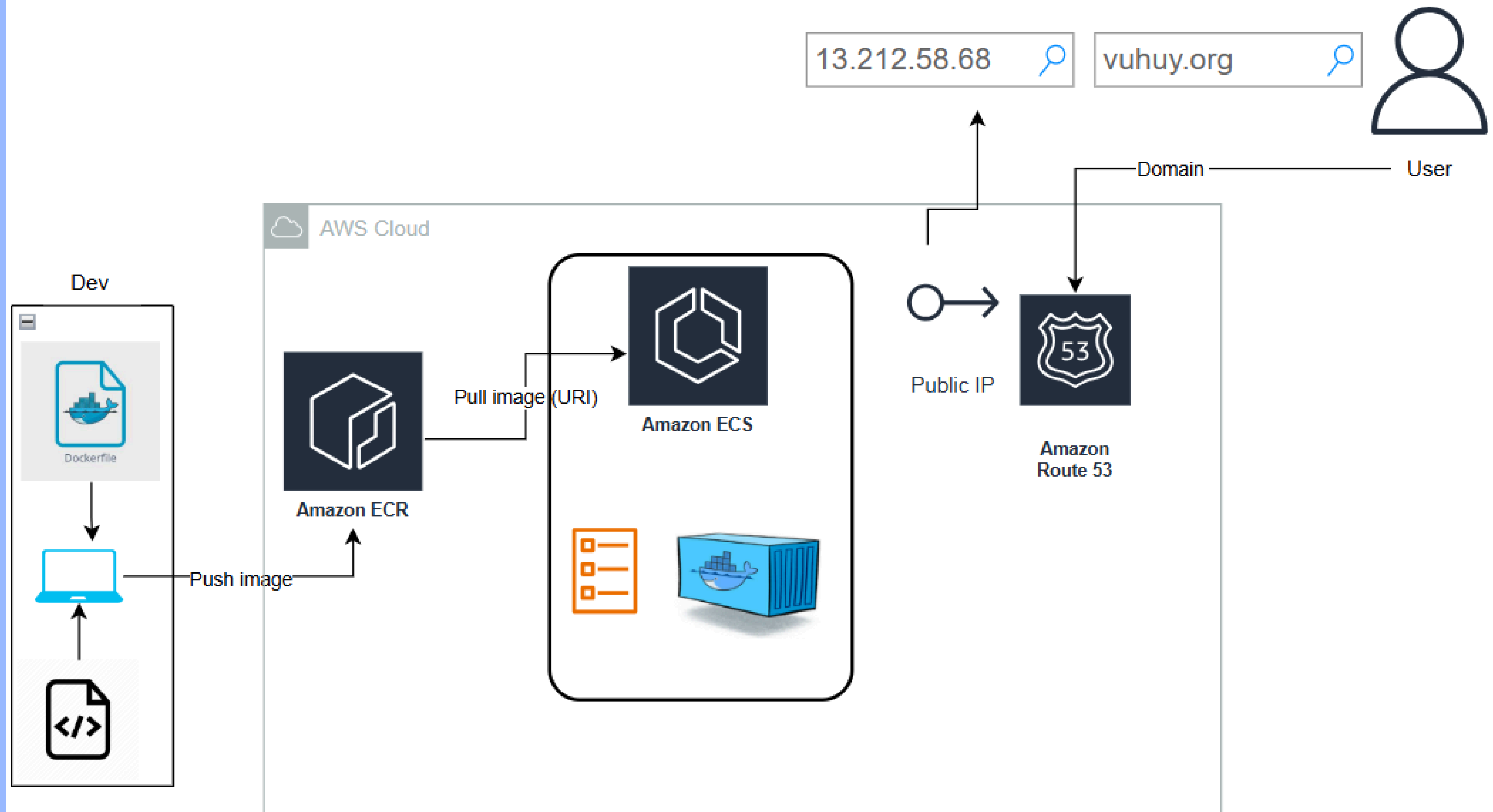
Task CPU

1024 units (1 vCPU)

Task memory

3072 MiB (3 GB)

System diagram



DEMO

Traffic Prediction Application

Select Model

Random Forest

Date

Time (HH:MM:SS AM/PM)

e.g. 02:30:00 PM

Car Count

Bike Count

Bus Count

Truck Count

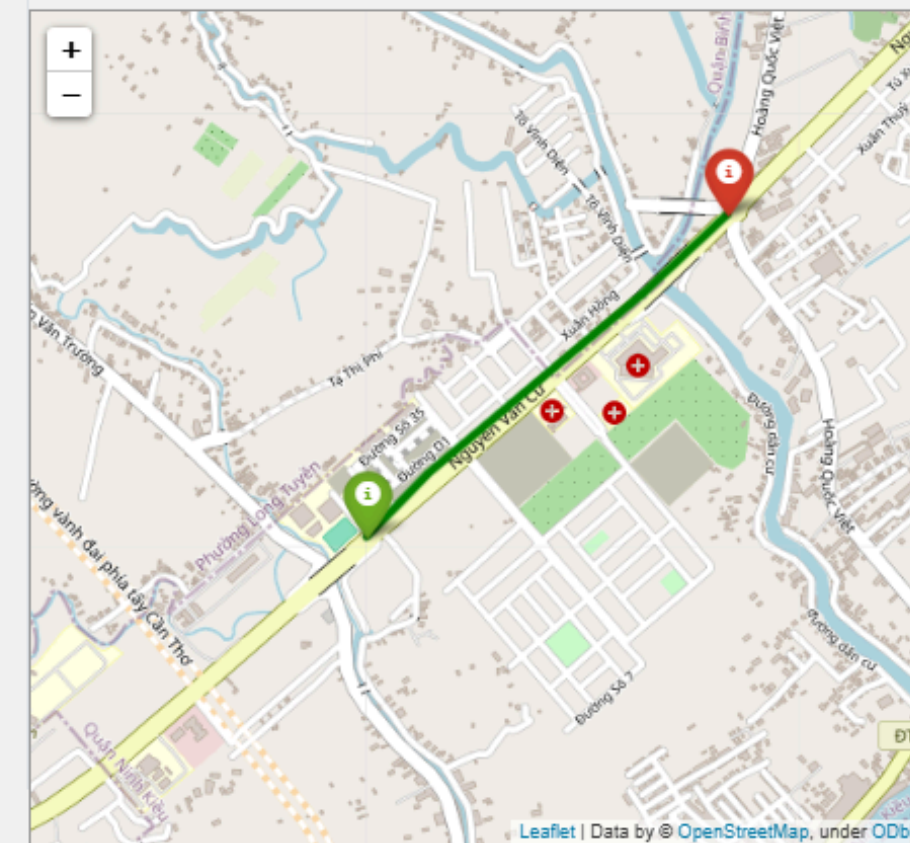
Total Vehicle Count

Day of the Week

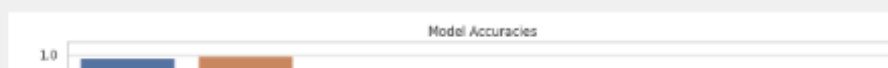
Monday

Predict

Traffic Map



Model Accuracies



Traffic Situation Distribution

