Faculty of Engineering & Technology

Department of Computer Science and Engineering

(Artificial Intelligence & Machine Learning)

Jain Global Campus, Kanakapura Taluk - 562112
Ramanagara District, Karnataka, India

2024-2025

(8th Semester)

A report on

Assignment Work

"Vehicle Count Prediction"

Submitted in fulfilment for the project work

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

(Artificial Intelligence and Machine Learning)

Submitted by

Atharva Gangodkar (21BTRCL016)

Under the guidance of

S Sivaram

Trainer

PhD. In Cybersecurity

Introduction

Traffic congestion is a growing concern in urban areas, and accurate vehicle count prediction can help in traffic management, resource allocation, and urban planning. This project aims to predict vehicle counts based on various time-related features using machine learning models.

Objectives

The goal of this project is to build a predictive model that can estimate the number of vehicles at a given time based on historical data. A graphical user interface (GUI) is implemented using Tkinter to allow users to input date and time details and obtain predictions using different machine learning models.

Technologies Used

- **Programming Language:** Python
- **Libraries:** Pandas, NumPy, Tkinter, Sklearn, XGBoost
- **Machine Learning Models:** Linear Regression, XGBoost, Lasso, Ridge
- **Data Processing:** StandardScaler for feature scaling
- **GUI Framework:** Tkinter for interactive input and result display.

Code - snippets

Data Preprocessing:

```
import pandas as pd

from sklearn.preprocessing import StandardScaler

data = pd.read_csv('VehicleCount.csv')

df = data.copy()

df['DateTime'] = pd.to_datetime(df['DateTime'])

df['Day'] = df['DateTime'].dt.day

df['Month'] = df['DateTime'].dt.month

df['Year'] = df['DateTime'].dt.year
```

```
df['Weekday'] = df['DateTime'].dt.weekday

df['Quarter'] = df['DateTime'].dt.quarter

df['DayOfYear'] = df['DateTime'].dt.dayofyear

df['Hour'] = df['DateTime'].dt.hour

df.drop(['DateTime'], axis=1, inplace=True)
```

Model Training:

```
from xgboost import XGBRegressor

from sklearn.linear_model import LinearRegression, Lasso, Ridge

scaler = StandardScaler()

x_scaled = scaler.fit_transform(df.drop(['Vehicles'], axis=1))

y = df['Vehicles']

models = {

    "Linear Regression": LinearRegression(),

    "XGBoost": XGBRegressor(),

    "Lasso": Lasso(),

    "Ridge": Ridge()

}

for model in models.values():

    model.fit(x_scaled, y)
```

Gui:

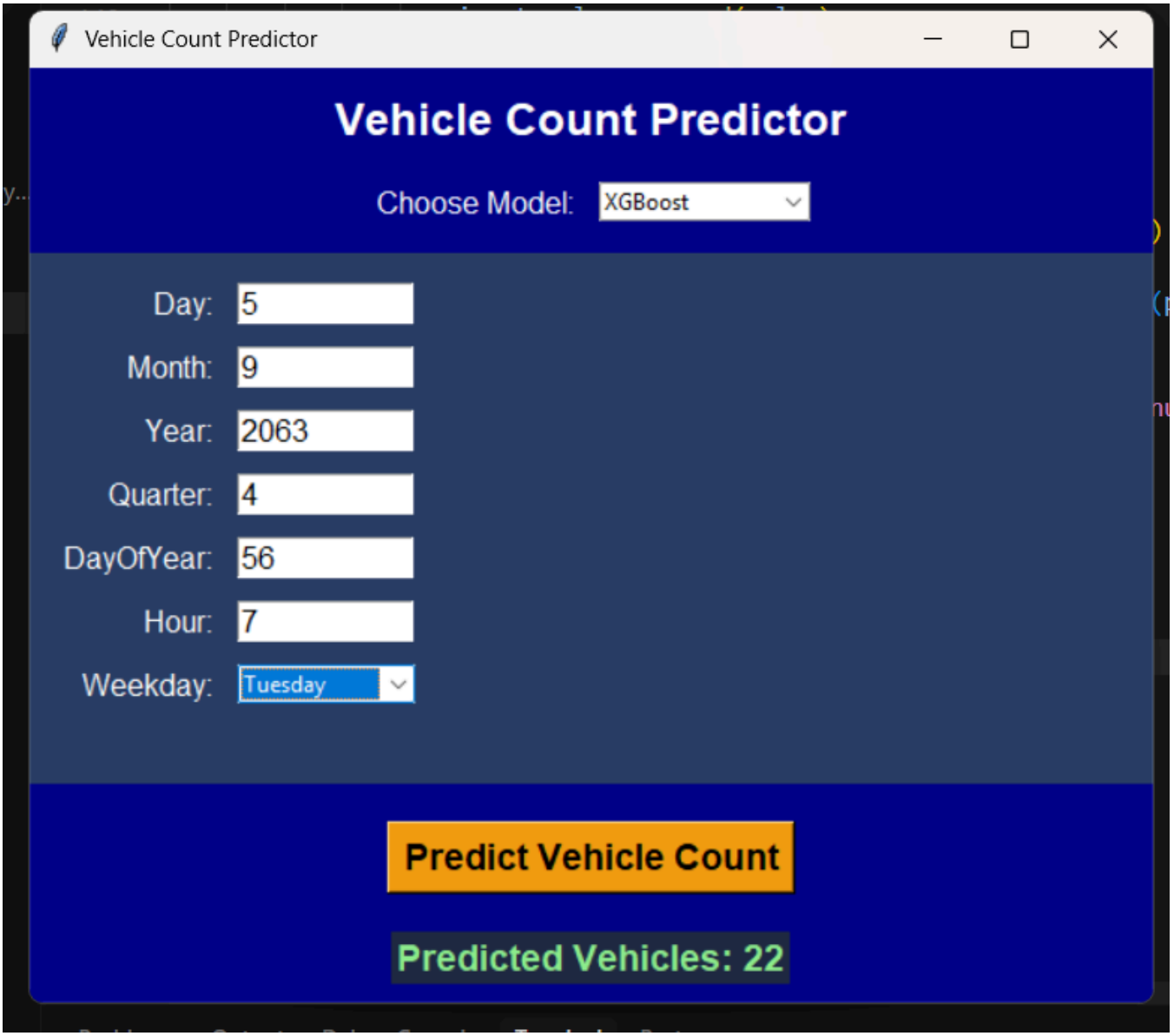
```
import tkinter as tk

from tkinter import ttk, messagebox

import numpy as np
```

```
class VehiclePredictorApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Vehicle Count Predictor")
        self.model_var = tk.StringVar(value="XGBoost")
        self.entries = {}
        fields = ["Day", "Month", "Year", "Quarter", "DayOfYear", "Hour", "Weekday"]
        for field in fields:
            self.entries[field] = tk.Entry(root)
            self.entries[field].pack()
        predict_btn = tk.Button(root, text="Predict", command=self.predict)
        predict_btn.pack()
        self.result_label = tk.Label(root, text="Result will be displayed here")
        self.result_label.pack()
    def predict(self):
        input_values = [float(self.entries[field].get()) for field in self.entries]
        input_scaled = scaler.transform([input_values])
        model = models[self.model_var.get()]
        prediction = model.predict(input_scaled)
        self.result_label.config(text=f"Predicted Vehicles: {int(prediction[0])}")
if __name__ == "__main__":
    root = tk.Tk()
    app = VehiclePredictorApp(root)
    root.mainloop()
```

Screenshots



Conclusion

This project successfully implements a vehicle count predictor using machine learning models. The user-friendly GUI allows for easy input and selection of different models for prediction. This tool can be extended further by incorporating real-time data sources and enhancing model accuracy using advanced techniques.