### **Road Accident Severity Prediction - Documentation**

#### Overview

This project aims to predict the severity of road accidents based on various input features using a machine learning model deployed with Flask. The model leverages Random Forest classification for accurate predictions. This document covers API details, model performance, setup instructions, and testing results.

#### 1. API Documentation

#### **Endpoints**

}

### 1. POST /predict (Predict Accident Severity)

```
Method: POST
Request Body (JSON):
{
 "num vehicles": 2,
 "time": 14,
 "road class": "A",
 "road_surface": "Wet / Damp",
 "lighting": "Darkness: street lights present and lit",
 "weather": "Fine without high winds",
 "casualty_severity": "Slight",
 "sex_of_casualty": "Male",
 "age_of_casualty": 35,
 "type_of_vehicle": "Car",
 "age_group": "60-69",
 "vehicle_group": "Two-Wheeled Vehicle"
}
Response (JSON):
 "severity": "Driver"
```

### 2. GET /health (Health Check)

• Method: GET

• Response:

•

• "status": "API is running"

• }

### 2. Model Performance Metrics

• Algorithm: Random Forest Classifier

• Accuracy: 60%

• Precision: 60%

• **Recall:** 99%

• **F1-Score**: 75%

• Class Imbalance Handling: SMOTE (Synthetic Minority Over-sampling Technique)

### 3. Setup Guide

### **Prerequisites:**

- Python 3.8+
- pip

### **Installation Steps:**

- 1. Clone the Repository:
- 2. git clone https://github.com/Abuhamida/road-accident-severity-prediction.git
- 3. cd road-accident-severity-prediction
- 4. Create a Virtual Environment:
- 5. python -m venv venv
- 6. source venv/bin/activate # Linux/macOS
- 7. venv\Scripts\activate # Windows
- 8. Install Dependencies:
- 9. pip install -r requirements.txt

## 10. Run the Flask Application:

11. python app.py

The app will be available at: http://127.0.0.1:5000/

### 4. Testing Results

# **Running Unit Tests**

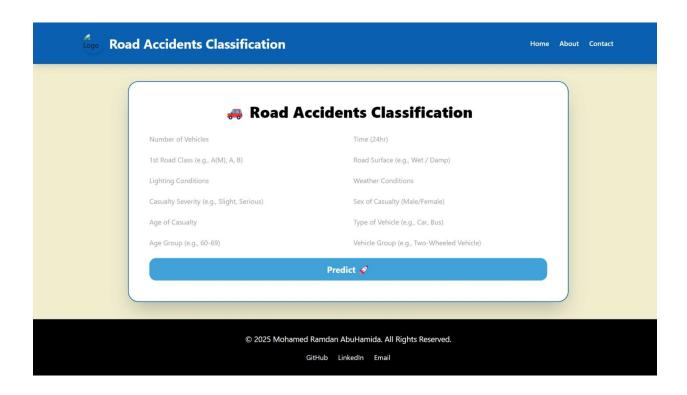
- Execute unit tests with:
- python -m unittest tests/test\_app.py
- Expected Output:
- •
- Ran 3 tests in 0.046s
- •
- OK

## **Key Functionalities Tested:**

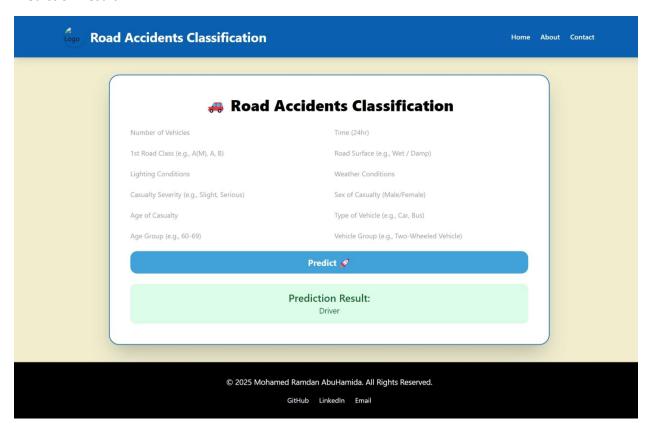
- API endpoint response validation.
- Model prediction accuracy.
- Edge cases and error handling.

### 5. Screenshots

**Homepage Interface:** 



### **Prediction Result:**



## 6. Troubleshooting

## **App Not Starting?**

- Ensure best\_model.pkl, label\_encoders.pkl, and scaler.pkl are in the root directory.
- Confirm Python version compatibility.

### **Test Failures?**

- Ensure form keys in test\_app.py match those in app.py.
- Check if the Flask app is running before executing tests.

### 7. Contact Information

• Author: Mohamed Ramdan AbuHamida

• **GitHub:** <u>Abuhamida</u>

• LinkedIn: Mohamed AbuHamida

• Email: mohamedabuhamida3@gamil.com