# **GROUP 8**

AMINA BASHIR – 674220

NICHOLAS BAHATI – 672949

IKIGU ELIZABETH – 674201

SYLVIA MACHARIA – 673922

# **Accident Severity Prediction Model Evaluation Report**

## **1. Objective:**

The primary objective of this project was to develop and evaluate a predictive model capable of classifying the severity of traffic accidents based on various factors, including weather conditions, road type, traffic density, and driver-related features. The target variable is the severity of accidents, categorized into three classes:

* **Class 0:** Low severity
* **Class 1:** Medium severity
* **Class 2:** High severity

The model's performance is evaluated using classification metrics such as **accuracy**, **precision**, **recall**, **F1-score**, and a **confusion matrix**.

## **2. Data Overview:**

The dataset contains the following columns:

* **Accident\_ID**: Unique identifier for each accident.
* **Date**: Date of the accident.
* **Time**: Time of the accident.
* **Severity**: Severity of the accident (target variable).
* **Weather\_Condition**: Weather condition at the time of the accident.
* **Road\_Type**: Type of road where the accident occurred.
* **Visibility**: Visibility during the accident.
* **Temperature\_C**: Temperature in Celsius.
* **Traffic\_Density**: Traffic density on the road.
* **Speed\_Limit**: Speed limit in the area of the accident.
* **Traffic\_Signal**: Presence of a traffic signal.
* **Vehicle\_Count**: Number of vehicles involved.
* **Driver\_Age**: Age of the driver.
* **Alcohol\_Involved**: Whether alcohol was involved.
* **County**: County where the accident occurred.
* **Road\_Name**: Name of the road.
* **Police\_Response\_Time**: Time taken for police to respond.
* **Driver\_Age\_Group**: Age group of the driver.

## **3. Methodology:**

The model used for prediction is a **Decision Tree Classifier**, a popular machine learning algorithm that works well for classification problems. The following steps were followed:

1. **Preprocessing**: Data cleaning, encoding categorical variables, and handling missing values.
2. **Model Training**: The data was split into training and testing sets (80% train, 20% test). A decision tree classifier was trained on the training set.
3. **Model Evaluation**: The model was evaluated using common classification metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

## **4. Model Evaluation Results:**

### **Accuracy:**

* **Accuracy**: 61%
* The model correctly predicted the accident severity in 61% of cases, indicating a moderately good performance. However, the model still has room for improvement.

### **Precision, Recall, and F1-Score:**

The following metrics were obtained from the classification report:

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1-Score** |
| **0** | 0.63 | 0.95 | 0.76 |
| **1** | 0.22 | 0.04 | 0.06 |
| **2** | 1.00 | 0.00 | 0.00 |

* **Class 0 (Low Severity)**: The model performed well with high precision (0.63) and recall (0.95), which indicates that it is effective in predicting low severity accidents.
* **Class 1 (Medium Severity)**: Precision and recall for medium severity accidents were low, suggesting the model struggles to detect medium severity accidents.
* **Class 2 (High Severity)**: Precision for high severity accidents was perfect (1.00), but recall was zero, meaning the model never predicted high severity accidents correctly when they actually occurred.

### **Macro Average:**

* **Precision**: 0.62
* **Recall**: 0.33
* **F1-Score**: 0.27
* These values reflect the overall performance, but the low recall and F1-score indicate challenges with predicting minority classes (medium and high severity).

### **Weighted Average:**

* **Precision**: 0.54
* **Recall**: 0.61
* **F1-Score**: 0.50
* The weighted average takes into account the class imbalance and gives a more balanced view of model performance across all classes.

### **Confusion Matrix:**

The confusion matrix visually shows the classification performance. The model tends to over-predict low severity accidents (Class 0) and under-predict medium (Class 1) and high severity (Class 2) accidents.

## **5. Key Findings:**

* **Model Strengths**: The model shows a strong ability to classify low severity accidents (Class 0) with high recall, meaning it successfully identifies most low-severity accidents.
* **Model Weaknesses**:  
  + **Medium Severity Accidents (Class 1)**: The model’s performance in predicting medium severity accidents is poor, with low precision, recall, and F1-score.
  + **High Severity Accidents (Class 2)**: Although the model predicts high severity accidents with perfect precision, it fails to identify any actual high severity accidents, as indicated by the zero recall.
* This suggests that the model may have **class imbalance** issues, where low-severity accidents dominate the dataset, and medium or high severity accidents are underrepresented.

## **6. Recommendations:**

1. **Address Class Imbalance**: Techniques like **SMOTE (Synthetic Minority Over-sampling Technique)** or **undersampling** could be used to balance the dataset and improve model performance, especially for minority classes.
2. **Hyperparameter Tuning**: Tune the hyperparameters of the decision tree model (e.g., max depth, min samples split) to prevent overfitting and improve its generalization ability.
3. **Feature Engineering**: Explore creating new features or interactions between existing features that might provide better signals for classifying medium and high severity accidents.
4. **Use Advanced Models**: Consider using ensemble models like **Random Forest** or **XGBoost** that can better handle class imbalance and offer improved performance.

## **7. Conclusion:**

The model provides a solid starting point for accident severity prediction but needs further refinement to handle medium and high severity accidents effectively. Addressing class imbalance and exploring more advanced models and techniques should improve performance.

**REQUIREMENTS**

1. MATPLOTLIB
2. SEABORN
3. PANDAS
4. NUMPY
5. RANDOM FORESTS
6. STATSMODEL
7. SKLEARN
8. SCIPYSTATS