**Dataset link: https://www.kaggle.com/datasets/saurabhshahane/road-traffic-accidents**

**1. Description of the Project:**

The goal of the project is to build a machine learning model that can classify the severity of road traffic accidents based on various features such as weather conditions, road surface conditions, light conditions, and more. The dataset used for this project contains information about different accidents, including their severity, as well as the aforementioned features. By training a classifier on this data, the project aims to provide a tool that can predict the severity of future accidents given their characteristics.

**2. List of Outputs with Detailed Descriptions:**

a) Trained Model: The project generates a trained machine learning model that can classify the severity of road traffic accidents. This model can be used to make predictions on new, unseen data.

b) Model Evaluation Metrics: The code provides various evaluation metrics, such as accuracy, precision, recall, and F1 score, to assess the performance of the trained model. These metrics help in understanding how well the model is performing in terms of classification accuracy and the trade-off between precision and recall.

c) Confusion Matrix: The code generates a confusion matrix, which provides a detailed breakdown of the predictions made by the model. It shows the number of true positives, true negatives, false positives, and false negatives, allowing for a deeper analysis of the model's performance.

**3. Main Output:**

The main output of this project is the trained machine learning model. This model can be used to predict the severity of road traffic accidents based on the provided features. Given a set of input features, the model will assign a class label to indicate the severity of the accident, such as minor, moderate, or severe.

**4. Detailed Instructions for Beginners to Run the Code:**

Step 1: Dataset Preparation

- Download the dataset using the provided link

- Place the dataset file in the same directory as the code file.

Step 2: Setting Up the Environment

- Install the required libraries mentioned in the code file. You can use pip or conda package managers to install the dependencies.

- Import the necessary libraries and modules in your Python environment.

Step 3: Loading the Data

- Modify the code to specify the correct file path and dataset name if you're using your own dataset.

- Run the code to load the dataset into memory.

Step 4: Data Preprocessing

- If needed, modify the code to perform any necessary data preprocessing steps, such as handling missing values, encoding categorical variables, or scaling numerical features.

Step 5: Splitting the Data

- Modify the code to adjust the train-test split ratio if desired. By default, the code uses a 70:30 train-test split.

- Run the code to split the data into training and testing sets.

Step 6: Model Training

- Modify the code to adjust the hyperparameters of the machine learning model if desired. For example, you can change the number of decision trees in the Random Forest classifier or modify the parameters of other classifiers.

- Run the code to train the model using the training data.

Step 7: Model Evaluation

- Once the model is trained, the code automatically evaluates its performance on the test data.

- Look for the evaluation metrics and the confusion matrix in the output. These provide insights into the model's performance.

Step 8: Making Predictions

- After evaluating the model, you can use it to make predictions on new, unseen data.

- Modify the code to provide the necessary features for the new data.

- Run the code to get the predicted severity class label for the new data.

Note: Make sure to read the code comments and documentation for any additional instructions or modifications specific to the code you are using.