**Project Overview: End-to-End Traffic Accident Prediction**

**1. Introduction**

The objective of this project was to develop a complete, end-to-end machine learning system to predict the likelihood of a traffic accident. Using a real-world dataset, the project followed a full MLOps (Machine Learning Operations) lifecycle, from initial data analysis and model training to deployment as a public-facing web application.

**2. Data Analysis and Preprocessing**

* **Initial Challenge:** The raw dataset contained numerous missing values and was composed of mixed data types, including categorical features (e.g., Weather, Road\_Condition) and numerical features (e.g., Speed\_Limit, Driver\_Age). It also suffered from a significant class imbalance, with far more "no accident" cases than "accident" cases.
* **Solution:** A robust data preprocessing pipeline was implemented to clean and transform the data.
  + **Handling Missing Values:** Missing numerical data was imputed with the **median**, a method that is resilient to outliers. Missing categorical data was filled with the **mode** (most frequent value).
  + **Feature Encoding:** Categorical features were converted into a numerical format suitable for machine learning models. **One-Hot Encoding** was applied to nominal features (e.g., Weather) to avoid misleading the model, while **Label Encoding** was used for ordinal features (e.g., Accident\_Severity).

**3. Model Training and Hybrid Logic**

* **Problem:** Due to the data's class imbalance, a standard model would be highly biased toward predicting the majority class ("no accident"). This would make the application unresponsive to changes in high-risk inputs.
* **Solution:** A powerful **XGBoost Classifier** was chosen for its high performance on tabular data. To overcome the model's bias, a **hybrid logic layer** was implemented in the application's backend. This system combines the model's prediction with a set of logical rules. For example, if a user inputs a combination of a high speed limit on an icy road with a driver under the influence of alcohol, the system overrides the model and explicitly returns a prediction of "Accident Likely." This approach ensures the application is both intelligent and responsive to critical inputs.

**4. Application Development and Containerization**

* **Challenge:** The trained model, a .pkl file, needed a user-friendly interface to be practical.
* **Solution:**
  + **Backend API:** A web service was built using the **Flask** framework in Python. This API includes a /predict endpoint that receives user data, preprocesses it correctly, and returns a prediction in JSON format.
  + **Frontend UI:** A single-page web interface was created with HTML, CSS (Tailwind CSS), and JavaScript. It provides an intuitive form for users to input data and displays the prediction result.
  + **Docker:** To ensure portability and a consistent environment, the entire application was containerized using **Docker**. A Dockerfile was created to package the application code, the trained model, and all Python dependencies into a single, isolated image.

**5. Cloud Deployment**

* **Challenge:** The local Docker container needed to be deployed to a public-facing server.
* **Solution:** **AWS Elastic Beanstalk** was used for deployment.
  + **Credentials and Security:** An **IAM user** was created with specific permissions, following the principle of least privilege. The AWS CLI and EB CLI were configured with these credentials.
  + **Deployment Workflow:** A Procfile and Dockerrun.aws.json file were created to instruct Elastic Beanstalk on how to run the Docker container.
  + **Final Deployment:** The eb init and eb create commands were used to automatically provision all necessary AWS resources (EC2 instance, load balancer, etc.) and deploy the application. The **security group** was configured to allow public access, resolving issues with accessing the application from different networks.

This project demonstrates the complete process of building a machine learning product, from a simple idea to a live, functional, and secure web service.