Week4: Deployment on Flask

Aya SDOUR 08-03-2025

1. Introduction

- **Purpose**: This report documents the deployment process of a machine learning model designed to predict airline safety improvements based on historical incident data.
- **Overview**: The project involves training a RandomForestClassifier on airline safety data and deploying it using a Flask web application to serve predictions.

2. Project Details

- **Name**: Aya SDOUR
- **Batch Code**: https://github.com/ay0788/data/tree/main/flask
- **Submission Date**: 2025-03-08
- **Submitted To**: Team Lead

3. Data Preparation

- **Data Loading**: The dataset `airline-safety.csv` is loaded using `pandas`. This dataset contains information about airline incidents and available seat kilometers.
- **Feature Engineering**:
  - Calculated `incident_rate_85_99` and `incident_rate_00_14` to normalize incident counts by available seat kilometers.
  - Created a binary label `safety_improved` to indicate whether the incident rate improved from 1985-1999 to 2000-2014.

4. Model Training

- **Feature Selection**: The features used for training are:
  - `incident_rate_85_99`: Incident rate from 1985 to 1999.
  - `fatal_accidents_85_99`: Number of fatal accidents from 1985 to 1999.

    - `fatalities_85_99`: Number of fatalities from 1985 to 1999.
    - `avail_seat_km_per_week`: Available seat kilometers per week.
- **Train-Test Split**: The data is split into 70% training and 30% testing sets using `train_test_split`.
- **Model Choice**: A `RandomForestClassifier` with 100 estimators is used for its robustness and ability to handle non-linear relationships.
- **Training Process**: The model is trained on the training set with a fixed random state for reproducibility.

5. Model Evaluation

- **Accuracy**: The model's accuracy on the test set is reported. For example, if the accuracy is 0.85, it means the model correctly predicts safety improvements 85% of the time.
- **Model Saving**: The trained model is saved to a file named `airline_safety_model.pkl` using `joblib` for later use in the Flask application.

6. Deployment

- **Flask Application**:
    - A Flask app is set up to serve the model. The app has two routes:
        - `/`: Renders the home page with a form to input data.
        - `/predict`: Accepts form data, makes a prediction using the model, and returns the result.
- **Prediction Logic**: The input data from the form is used to make a prediction. If the prediction is 1, it indicates that safety is likely to improve; otherwise, it is not.

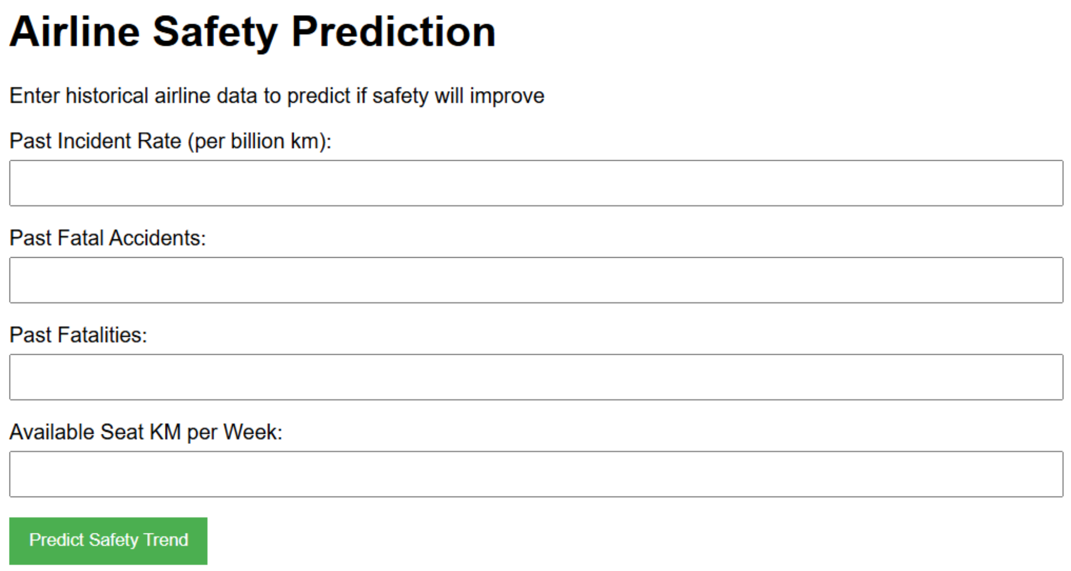
7. Snapshots

- **Data Loading and Feature Engineering**: Screenshot of the dataset after loading and feature engineering steps.
- **Model Training and Evaluation**: Code snippet or output showing the training process and accuracy.
- **Flask App Setup**: Screenshot of the Flask app running and the home page.
- **Prediction Endpoint**: Example of a prediction result displayed on the web page.

8. Conclusion

- **Summary**: The deployment process involved training a RandomForest model on airline safety data and deploying it using Flask.

- **Challenges**: Any challenges faced during deployment, such as data preprocessing or model serving issues.
- **Next Steps**: Recommendations for future improvements, such as model optimization or adding more features.



Airline Safety Prediction

Enter historical airline data to predict if safety will improve

Past Incident Rate (per billion km):

Past Fatal Accidents:

Past Fatalities:

Available Seat KM per Week: