Design and Deployment of an AI-Powered Predictive System

# 📌 Project Overview

Project Title: Design and Deployment of an AI-Powered Predictive System

Project Question: How can AI models be developed and deployed in real-world applications to support decision-making and improve task automation through a simple web interface?

Use Case: Fraud Detection – Identifying fraudulent transactions using payment data patterns.

# 🧪 Model Summary

The following supervised ML models were trained and evaluated:

| Model | Precision | Recall | F1 Score | ROC AUC |

|--------------------|-----------|--------|----------|---------|

| Logistic Regression| 0.91 | 0.87 | 0.89 | 0.92 |

| Random Forest | 0.96 | 0.93 | 0.94 | 0.98 |

| XGBoost | 0.97 | 0.94 | 0.95 | 0.99 |

| Neural Network | 0.96 | 0.92 | 0.94 | 0.97 |

# 📂 Download Dataset & Resources

To keep the repo clean and under GitHub’s size limit, datasets and large model files are stored externally:

Download via Google Drive: https://bit.ly/dsa\_ai\_ml

Includes:

- train\_transaction.csv

- smote\_balanced\_data.csv

- model.pkl, scaler.pkl, feature\_columns.pkl

- Flask app files (for offline prediction testing)

# ⚙️ Installation & Setup

Clone the Repository:

git clone https://github.com/btolawoyin/dsa\_ai\_ml.git

cd dsa\_ai\_ml

Create and Activate Virtual Environment (Windows):

python -m venv venv

venv\Scripts\activate

Install Dependencies:

pip install -r requirements.txt

# 🚀 Run Flask App Locally

python app.py

Then visit: http://127.0.0.1:5000/ in your browser.

# 🌐 Deployment (Heroku-ready)

1. Ensure Procfile and requirements.txt are present.

2. Use the following commands:

heroku login

heroku create your-app-name

git push heroku main

# 💡 Reflection & Future Improvements

✅ What Worked:

- SMOTE effectively addressed class imbalance.

- XGBoost gave high recall and AUC.

- Clean and interactive Flask interface for predictions.

⚠️ What Didn’t:

- Neural Networks had longer training times.

- Some input features required extra encoding effort.

🔮 Future Plans:

- Integrate real-time transaction streaming.

- Use more sophisticated anomaly detection techniques.

- Containerise the app with Docker for wider deployment.

# 🧑‍💻 Tech Stack

- Python 3.10

- Flask

- Pandas, NumPy, Scikit-learn, XGBoost

- Seaborn, Matplotlib

- SMOTE (Imbalanced-learn)

# 📧 Contact

Built by Bukky Tolawoyin (https://github.com/btolawoyin)

For DSA AI/ML Course Final Project (2025)