**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:

* Logistic Regression
* Random Forest
* XGBoost
* Neural Network (MLP Classifier)

**Performance Metrics:**

* Precision
* Recall
* F1 Score
* ROC AUC

Each model was assessed to determine predictive accuracy on imbalanced transaction datasets.

**Data & Resources**

Download datasets and resources here: [**Google Drive - DSA AI/ML Resources**](https://bit.ly/dsa_ai_ml)

**Files Available:**

* train\_transaction.csv (Raw dataset)
* smote\_balanced\_data.csv (Balanced dataset using SMOTE)
* model.pkl (Trained model)
* scaler.pkl (Feature scaler used in training)
* feature\_columns.pkl (Selected input features)
* Flask app files (app.py, templates/, etc.)

**Note:** Large datasets are not committed to GitHub due to file size limits.

**Setup Instructions (Windows 11 – VS Code)**

1. **Clone the repository:**git clone https://github.com/btolawoyin/dsa\_ai\_ml.git

cd dsa\_ai\_ml

1. **Create a virtual environment:**

python -m venv venv

venv\Scripts\activate

1. **Install all dependencies:**

pip install -r requirements.txt

**Running the Flask Web App**

To start the application locally: python app.py

Then open your browser and go to: <http://127.0.0.1:5000/>

You’ll see a form to enter transaction data and get a fraud prediction.

**Reflection**

**What Worked**

* SMOTE helped address extreme class imbalance.
* XGBoost produced superior model performance.
* Flask UI allowed intuitive interaction with the ML model.

**What Didn’t**

* Neural Networks took excessive training time.
* GitHub push was blocked due to large files.

**Future Improvements**

* Integrate streaming API for real-time fraud prediction.
* Dockerise and orchestrate with Kubernetes.
* Integrate alert systems for high-risk transactions.

**Tools Used**

* **Programming Language:** Python 3.10
* **Libraries:**
  + Pandas, NumPy, Scikit-learn
  + XGBoost, imblearn
  + Flask, Seaborn, Matplotlib
* **Version Control:** Git, GitHub
* **IDE:** Visual Studio Code on Windows 11

**📬 Author**

**Name:** Bukky Olawoyin  
**GitHub:** [github.com/btolawoyin/dsa\_ai\_ml](https://github.com/btolawoyin/dsa_ai_ml)   
**Project:** DSA Projects - Artificial Intelligence