Task 8: Feature Engineering and Model Tuning

This task focuses on **enhancing model performance** through **feature engineering** and **hyperparameter tuning**, improving its ability to detect fraudulent transactions.

📌 Section 1: Feature Engineering & Model Tuning

🎯 Objective

🔹 Improve model performance through **feature engineering** and **hyperparameter tuning**.

🛠️ Tasks to Perform

✅ **Feature Engineering**:

- Create **new features** (e.g., total score from subject scores in a student dataset).

✅ **Hyperparameter Tuning**:

- Use **GridSearchCV** to optimize model parameters.

📌 **Deliverable**: Improved **model accuracy and performance**.

📌 Section 2: Fraud Detection with Decision Trees

🎯 Objective

🔹 Detect fraudulent transactions based on patterns in **financial data**.

🛠️ Project Steps

1️⃣ Dataset Selection

📌 **Dataset:** `fraud_detection.csv`
📌 **Columns:**

- **Transaction ID**
- **Amount**
- **Type** (e.g., credit/debit)

- **Is Fraud** (1 = Fraud, 0 = Legitimate)

2️⃣ Tasks to Perform

✅ 1. Load & Preprocess the Dataset

- Read the data and **inspect missing values**.
- Convert **categorical variables** (e.g., transaction type) using **label encoding**.

✅ 2. Feature Engineering

- Create **new derived features** from transaction data.

✅ 3. Train a Decision Tree Classifier

- Split the dataset into **training and testing sets**.
- Train a **Decision Tree model** to classify transactions.

✅ 4. Evaluate Model Performance

- Use **precision, recall, and F1-score** to measure effectiveness.
- Identify potential **improvements** in fraud detection.

📌 Deliverables

📌 **1. Decision Tree Model** trained for fraud detection.
📌 **2. Evaluation Metrics** (Precision, Recall, F1-Score).
📌 **3. Recommendations** on how to improve fraud detection accuracy.

Would you like recommendations for additional **fraud detection techniques**, such as **Random Forest** or **Anomaly Detection**? 🚀

Deadline Compliance

- **Restriction**: **Submit the project within 7 days** from the start date.
- **Reason**: Meeting deadlines is crucial in the real-world software development environment. This restriction helps students practice **time management** and **task prioritization**. In professional settings, tight deadlines are often the norm, and learning to meet them without compromising quality is an essential skill.

- **Learning Outcome**: Students will learn to manage their time effectively, complete projects under pressure, and **deliver results on time**, which are all important skills in the workplace.