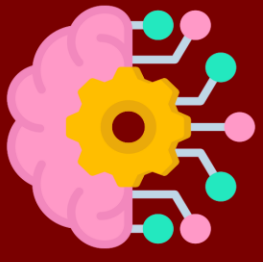
FRAUD DETECTION END TO END PROJECT



Latreche Sara -code213

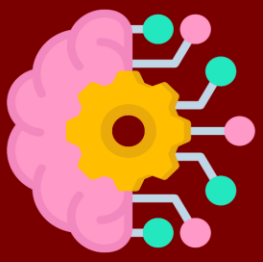
Project Context and Objective

Role: Junior Data Scientist (Fraud Operations) **Location:** European Fintech Hub **Objective:** Protecting the integrity of 280,000+ daily transactions.

The Context: The Silent Battle

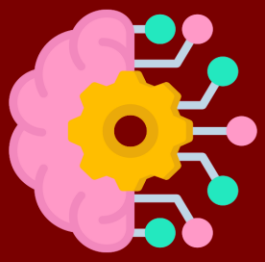
Welcome to the team. You've joined us at a critical time. In the digital economy, trust is our most valuable currency. Every time a customer swipes their card, they trust us to protect their hard-earned money. However, fraud is a sophisticated, moving target. In September 2013, our systems recorded two days of transactions from European cardholders. Out of nearly **285,000 transactions**, only **492** were fraudulent.

While that sounds small, those 492 transactions represent a massive breach of security and a significant financial loss. Our goal isn't just to find the "needle in the haystack"—it's to find the needle before the customer even knows it's there.



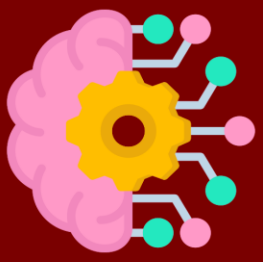
Step 1: Environment & Dataset Exploration

- **Task 1.1**: Set up your environment by installing `imbalanced-learn` and importing the necessary libraries.
- **Task 1.2**: Load the dataset and visualize the class distribution.
- **Question**: What is the percentage of fraudulent transactions? Why does this make standard "Accuracy" a deceptive metric for this project?



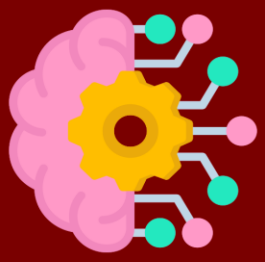
Step 2: Baseline Modeling (Logistic Regression)

- **Task 2.1**: Perform feature scaling using StandardScaler.
- **Task 2.2**: Train a standard LogisticRegression model and generate a classification_report.
- **Question**: How many fraud cases (Class 1) were actually caught? What do you observe about the Recall score?



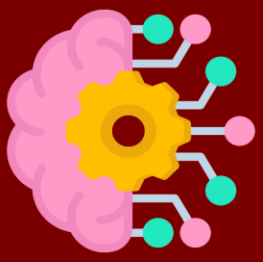
Step 3: Refining the Baseline

- **Task 3.1**: Re-train the model using the class_weight='balanced' parameter.
- **Task 3.2**: Plot the Precision-Recall curve.
- **Task 3.3**: Iterate through different probability thresholds .
- **Question**: How does changing the threshold affect the trade-off between Precision and Recall?



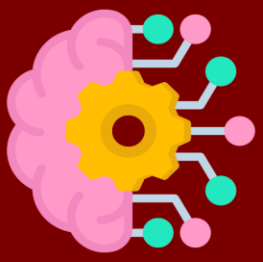
Step 4: Deep Learning Implementation (MLP)

- **Task 4.1**: Use the MLPClassifier with hidden layers set to (64, 32).
- **Task 4.2**: Evaluate this initial neural network.
- **Question**: Without resampling, what happened to the Recall for the fraud class? Did the model "learn" anything about fraud?



• Step 5: Advanced Resampling (SMOTE)

- **Task 5.1**: Apply the **SMOTE** oversampling strategy to your training data.
- **Task 5.2**: Re-train the `MLPClassifier` on this new balanced data.
- **Question**: How does the MLP performance change after it is exposed to synthetic fraud samples?



• Step 6: Operational Optimization

•**Task 6.1**: Optimize the decision threshold for your MLP+SMOTE model.
•**Final Analysis**: Compare the **PR-AUC** and **Confusion Matrix** of the Balanced Logistic Regression versus the Optimized MLP.
•**Question**: Which model provides the best "Operational Efficiency" for a bank (high detection with fewer false alarms)?