# Final Report - Fraud Detection (E-commerce + Bank)

## Final Report - End-to-End Fraud Detection (E-commerce + Bank)

### *1) Executive summary*

This project built an end-to-end fraud detection workflow for Adey Innovations Inc. covering:

- **E-commerce fraud** with **IP->country geolocation** enrichment and behavior/time features
- **Bank card fraud** with anonymized PCA features

The solution prioritizes imbalanced-learning best practices: stratified splits, train-only resampling, and AUC■PR/F1 evaluation.

### *2) Data understanding and business context*

Fraud detection is highly imbalanced. Operationally:

- False positives reduce customer trust and increase support cost
- False negatives directly increase fraud losses

Therefore, we evaluate models beyond accuracy using **AUC■PR**, **F1**, and confusion matrices.

### *3) Task 1 - Data preprocessing & feature engineering*

#### Cleaning

- Removed duplicates
- Fixed data types (timestamps/numerics)
- Managed missing values with practical imputations and safe drops for critical fields

#### Geolocation integration (Fraud_Data)

- Converted IP to integer
- Range-joined into `IpAddress_to_Country.csv`
- Assigned missing/invalid IPs to **Unknown**

#### Features (Fraud_Data)

- Time: `hour_of_day`, `day_of_week`, `time_since_signup_sec`
- Velocity: `user_txn_count_1h`, `user_txn_count_24h`
- Aggregates: user/device counts

#### Transformations

- Scaling: `StandardScaler`
- Encoding: `OneHotEncoder`
- Imbalance handling: **SMOTE on training data only**

## *4) Task 2 - Modeling & evaluation*

Models:

- Baseline: Logistic Regression (interpretable)
- Ensemble: Random Forest

Validation:

- Stratified K-Fold CV (k=5)
- Test evaluation with AUC■PR / F1 / confusion matrix

Outputs:

- Metrics: `reports/task2_*_results.json`
- Models: `models/task2_*_*.joblib`

## *5) Task 3 - Explainability (SHAP)*

Explainability was added via SHAP:

- Global: SHAP beeswarm (top drivers)
- Local: SHAP waterfall for:

- TP (true fraud caught)

- FP (legitimate flagged)

- FN (missed fraud)

Notebook:

- `notebooks/shap-explainability.ipynb`

## *6) Business recommendations (examples)*

Use SHAP insights to propose operational rules. Example recommendations:

1) **High velocity shortly after signup**:

- If transactions occur within a short time after signup and velocity spikes, require step-up verification (OTP/2FA).

2) **Country-risk based monitoring**:

- Increase monitoring for countries that show higher fraud rates in geolocation analysis.

3) **Device/user mismatch patterns**:

- Flag devices associated with many distinct users or users switching devices unusually often for additional checks.

These should be tuned to keep false positives manageable while reducing fraud loss.

## *7) How to reproduce*

Install base deps:

pip install -r requirements.txt

Task 1:

python -m scripts.task1_preprocess --dataset all

Task 2:

python -m scripts.task2_train --dataset all

Task 3:

pip install -r requirements-task3.txt

Then open:

- `notebooks/shap-explainability.ipynb`

## Appendix: Auto-generated metrics

```
{
  "class_counts": {
    "fraud_class_counts": {
      "0": 136961,
      "1": 14151
    },
    "creditcard_class_counts": {
      "0": 284315,
      "1": 492
    }
  },
  "task2_fraud_results_present": true,
  "task2_creditcard_results_present": false,
  "task2_fraud_summary": {
    "logreg": {
      "cv": {
        "n_splits": 5,
        "auc_pr_mean": 0.680561320840885,
        "auc_pr_std": 0.01743252563690526,
        "f1_mean": 0.6181296920115373,
        "f1_std": 0.015928932504549475
      },
      "test": {
        "auc_pr": 0.6914262594457254,
        "f1": 0.6268656716417911,
        "confusion_matrix": [
          [
            5171,
            294
          ],
          [
            181,
            399
          ]
        ]
      },
      "model_path": "models/task2_fraud_logreg.joblib"
    },
    "random_forest": {
      "cv": {
        "n_splits": 5,
        "auc_pr_mean": 0.7153496739305094,
        "auc_pr_std": 0.013879189743560342,
        "f1_mean": 0.6974448812398387,
        "f1_std": 0.01618300894779367
      },
      "test": {
        "auc_pr": 0.7309467984627795,
        "f1": 0.7021943573667712,
        "confusion_matrix": [
          [
            5424,
            41
          ],
          [
            244,
            336
          ]
        ]
      },
      "model_path": "models/task2_fraud_random_forest.joblib"
    }
  },
  "task2_creditcard_summary": {},
  "note": "If Task 2 results are missing, run: python -m scripts.task2_train --dataset all"
}
```