Interim-1 Report (Task 1) - Fraud Detection

Interim-1 Report - Task 1 (Data Analysis & Preprocessing)

Project overview

Adey Innovations Inc. needs robust fraud detection for:

• **E-commerce transactions** (`Fraud_Data.csv`) enriched with **IP->country** geolocation
• **Bank card transactions** (`creditcard.csv`) with PCA-derived features

Fraud detection is a high-stakes **imbalanced classification** problem, where:

• **False positives** harm customer experience (legitimate users blocked)
• **False negatives** cause direct financial loss (fraud missed)

Data sources (provided by the project)

• `Fraud_Data.csv` (target: `class`, where 1=fraud)
• `IpAddress_to_Country.csv` (IP range -> country)
• `creditcard.csv` (target: `Class`, where 1=fraud)

Repository structure

The repository is organized as required:

• `data/raw/`: raw CSV files (gitignored)
• `data/processed/`: generated artifacts (gitignored)
• `src/`: reusable preprocessing + feature engineering code
• `notebooks/`: EDA and feature engineering notebooks
• `scripts/`: runnable entrypoints

Task 1 - Cleaning and preprocessing

Data cleaning
Applied to both datasets:

• Removed duplicates
• Corrected data types (timestamps and numerics)
• Handled missing values with safe defaults:
- Drop rows missing essential fields (targets / timestamps)
- Median imputation for numeric fields where appropriate
- "Unknown" for missing categorical values

Geolocation integration (Fraud_Data)

To enrich e-commerce transactions, IP addresses were mapped to countries:

• IP addresses converted to integer format
• Range-based join using `lower_bound_ip_address` / `upper_bound_ip_address`
• Missing/invalid IPs safely labeled **Unknown**

This enables country-level fraud analysis and country-aware features.

Feature engineering (Fraud_Data)

Engineered features aligned with fraud patterns:

• **Time-based features**:
- `hour_of_day`
- `day_of_week`
- `time_since_signup_sec` (purchase_time - signup_time)
• **Transaction velocity**:
- `user_txn_count_1h`
- `user_txn_count_24h`
• **Behavior aggregates**:
- `user_total_txns`
- `user_unique_devices`
- `device_unique_users`

Data transformation

• Numeric scaling: `StandardScaler`
• Categorical encoding: `OneHotEncoder(handle_unknown="ignore")` (Fraud_Data)

Class imbalance handling (train only)

To address severe imbalance:

• **SMOTE oversampling** applied to **training data only**
• Train/test split is stratified to preserve imbalance in evaluation

How to reproduce Task 1

1) Put raw data in:

• `data/raw/Fraud_Data.csv`
• `data/raw/IpAddress_to_Country.csv`
• `data/raw/creditcard.csv`

2) Run preprocessing:

python -m scripts.task1_preprocess --dataset all

3) Outputs are written to `data/processed/`, including:

• Engineered datasets (`*.parquet`)

• Preprocessors (`*_preprocessor.joblib`)

• Train/test matrices (`*.npz` / `*.npy`)

• Metadata (`*_task1_metadata.json`) including class distribution before/after SMOTE

EDA notebooks

• `notebooks/eda-fraud-data.ipynb`

• `notebooks/eda-creditcard.ipynb`

• `notebooks/feature-engineering.ipynb`

Key takeaway

Task 1 produced **clean, feature-rich, reproducible** datasets suitable for imbalanced modeling, including geolocation enrichment and time/velocity-based fraud signals.

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"
}
```