

Interim-2 Report (Task 2) - Fraud Detection

Interim-2 Report - Task 2 (Model Building & Training)

Goal

Build and evaluate fraud detection models for both datasets using metrics appropriate for class imbalance:

- **AUC-PR** (Average Precision)
- **F1-score**
- **Confusion Matrix**

Data preparation

- Stratified train/test split (preserves fraud rate)
- Preprocessing applied inside the modeling pipeline
- **SMOTE applied only on training folds** (via `imblearn.Pipeline`)

Models trained

For each dataset:

- 1) **Baseline (interpretable)**: Logistic Regression (`class_weight="balanced"`)
- 2) **Ensemble**: Random Forest (`class_weight="balanced_subsample"`)

Cross-validation

- **Stratified K-Fold** with **k=5**
- Report **mean ± std** for AUC-PR and F1 on training folds

Reproducibility

Run Task 2:

```
python -m scripts.task2_train --dataset all
```

Artifacts:

- `models/task2__.joblib` (full pipeline)
- `reports/task2__results.json` (metrics summary)

Results summary

After running Task 2, consult:

- `reports/task2_fraud_results.json`

- `reports/task2_creditcard_results.json`

These files include:

- CV mean/std (AUC-PR, F1)
- Test-set AUC-PR, F1, confusion matrix

Model selection (business justification)

For fraud detection, model choice balances:

- **Higher AUC-PR** (better ranking of rare fraud cases)
- **F1** trade-off (precision/recall balance)
- Interpretability needs (baseline logistic regression remains valuable for explanation and debugging)

In practice, the ensemble often improves recall/precision on rare positives, but the baseline provides a transparent reference for stakeholders.

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": 0.7153496739305094
    }
  }
}
```

```
"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"
}
```