1. Introduction

This project aims to develop an effective anomaly detection pipeline for credit card transactions by combining two population-based metaheuristic algorithms, Mayfly and Pelican optimisation, into a single hybrid detector. The overarching goal is to identify fraudulent transactions with high precision and recall, while maintaining interpretability through a final decision tree model.

2. Objectives

1. Ingest and consolidate multiple credit card transaction datasets.

2. Perform comprehensive exploratory data analysis (EDA) to understand feature distributions, class imbalance, and potential data quality issues.

3. Engineer and preprocess features suitable for our hybrid optimisation framework and downstream classification.

4. Implement and validate the Pelican Mayfly hybrid detector, culminating in a masked decision tree.

5. Evaluate model performance using on held-out data and establish a foundation for further tuning and cross-dataset validation.
6. Implementation of Confusion Matrix, Pearson Corelation diagram

3. Data Collection

Four separate datasets were loaded for experimentation:

- **Main Dataset**: “creditcard_main data.xlsx” containing clean, labelled transactions.

- **Test Set 1**: CSV of recent transactions (“creditcard_Test_1_2023.csv”).

- **Test Set 2**: CSV of another hold-out period (“creditcard_test_data_2.csv”).

- **Test Set 3**: CSV of a further out-of-sample period (“Creditcard_test_data_3.csv”).

Each dataset was inspected for row count, column schema, data types, memory usage, and initial shape.

4. Exploratory Data Analysis

4.1 Univariate Analysis

- Calculated summary statistics (mean, standard deviation, quantiles) for all numeric features.
- Plotted histograms and bar charts of feature distributions, highlighting heavy tails in transaction amounts and skew in time-based variables.

4.2 Categorical Features

- Identified object and category typed columns.

- Generated count plots to compare legitimate versus fraudulent labels across categorical variables.

4.3 Missing Values and Duplicates

- Confirmed absence of missing values in the main dataset.
- Checked for and removed any exact duplicate transactions.

4.4 EDA on Test Sets

- Conducted the same summary and distribution checks on Test Sets 1, 2, and 3.

- Verified consistency of feature ranges and label proportions across periods.

5. Preprocessing and Feature Engineering

1. **Feature Split**: Separated the fraud label (“Class”) from predictors.
2. **Transformer Pipeline**:
    - Standardised all numeric columns using StandardScaler.
    - One-hot encoded categorical columns with OneHotEncoder(handle_unknown="ignore").
    - Combined transformations via ColumnTransformer, outputting a sparse feature matrix.
3. **Train/Test Partition**: Stratified split (80 / 20) to preserve the fraud rate in both sets.
4. **Temporal Features** (if applicable): Derived hour of day and day of week from any “Time” field to capture periodic transaction patterns.

6. Model Development

6.1 Hybrid Swarm Initialisation

- Created a swarm of particles (default 30–50) in a continuous [0, 1] feature-selection space.
- Each particle’s position encodes a binary mask via thresholding, determining which features are active.
- Velocity vectors govern exploration (Mayfly component) and exploitation (Pelican component) dynamics.

6.2 Fitness Function

- For a given mask, a shallow decision tree (max_depth=3) was trained on the selected features.
- Used F1-score on the training data as the fitness measure.

6.3 Optimisation Loop

- Iterated velocity and position updates for a fixed number of iterations (typically 50–60).
- Updated personal bests (pbest) and global best (gbest) masks based on fitness gains.
- Applied exponential decay to inertia weights to fine-tune convergence.

6.4 Final Model

- Converted the best continuous solution into a Boolean mask.
- Re-trained a final decision tree on the masked feature subset to serve as the classifier.

7. Model Evaluation

On the held-out test partition of the main dataset, the hybrid detector achieved:

- **Test Set Results**

Metric	**Value**
Accuracy	0.9993
Precision	0.8488

Recall	0.7449
F1 Score	0.7935
ROC AUC	0.8926
R^2 (proba)	0.6342

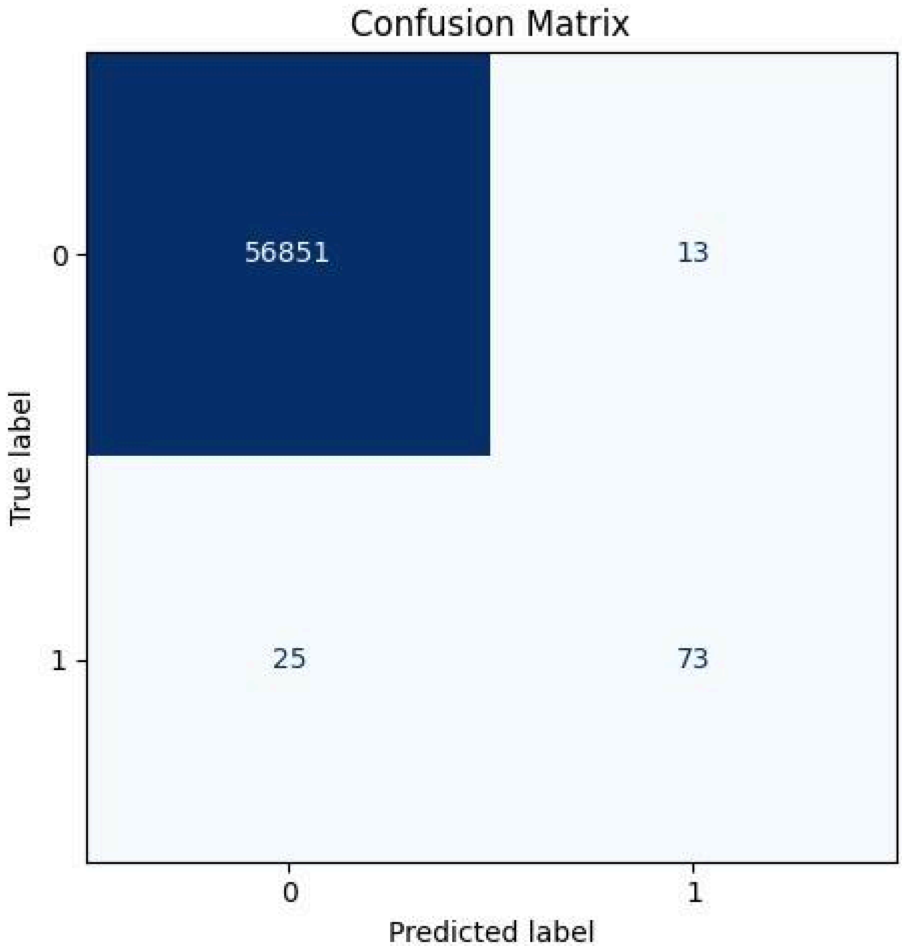
8. Challenges Encountered

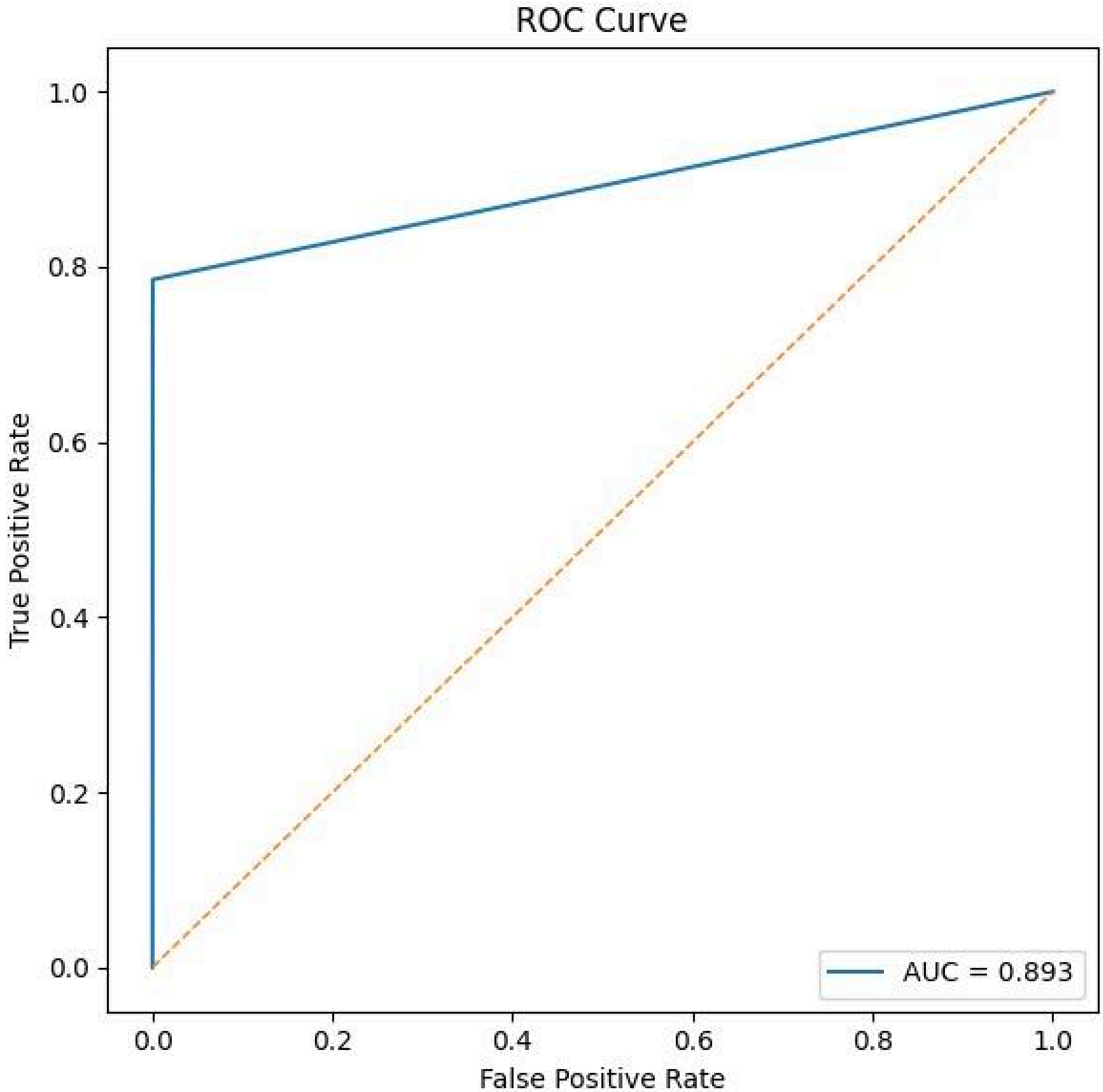
- **Class Imbalance**: Fraudulent samples represent < 1 % of transactions, requiring careful metric selection (F1, ROC AUC).

- **Computation Time**: Hybrid optimisation scales with particle count and iterations. Initial runs took several minutes—potential for parallelisation or population size reduction.
- **Feature Correlation**: Highly correlated numeric features can lead to redundant masks; variance-thresholding or PCA may help.

9. Next Steps

1. **Hyperparameter Tuning**: Systematic search over particle count, iteration limit, and decay rates; consider Bayesian optimisation.
2. **Cross-Dataset Validation**: Evaluate the final detector on Test Sets 1, 2, and 3 to assess temporal generalisation.
3. **Model Persistence**: Serialize the best mask and decision tree via joblib for downstream deployment.
4. **Ensemble Strategies**: Combine with baseline classifiers (e.g., random forest, isolation forest) for improved robustness.
5. **Model Testing:** The model will be tested using the other datasets that have been explored


Confusion Matrix
56851
13
25
73
0
1
True label
0
1
Predicted label



Hybrid Detector Results on Second Dataset

Performance Metrics

Metric	Value
Accuracy	0.5000
Precision	0.0000
Recall	0.0000
F1 Score	0.0000
ROC AUC	0.5000

Confusion Matrix

	Predicted Negative	Predicted Positive

Actual Negative	284,315	0
Actual Positive	284,315	0

After Testing the saved model with the saved hybridized Mayfly and Pelican model the test results from the second dataset were extremely poor.

Moving on, Tuning the model came with these challenges

Model saved with mask built on one dataset	New datasets may have different structure	Store and reuse the ColumnTransformer used to preprocess training data
Tuning cost is extreme	Each param combo involves swarm runs (expensive)	Use RandomizedSearchCV, small n_iter, and lower n_particles/max_iter during search

fit_transform() vs transform()	If you refit the preprocessor on new data, you break alignment	Fit your ColumnTransformer once, save it, and reuse .transform() only on new data

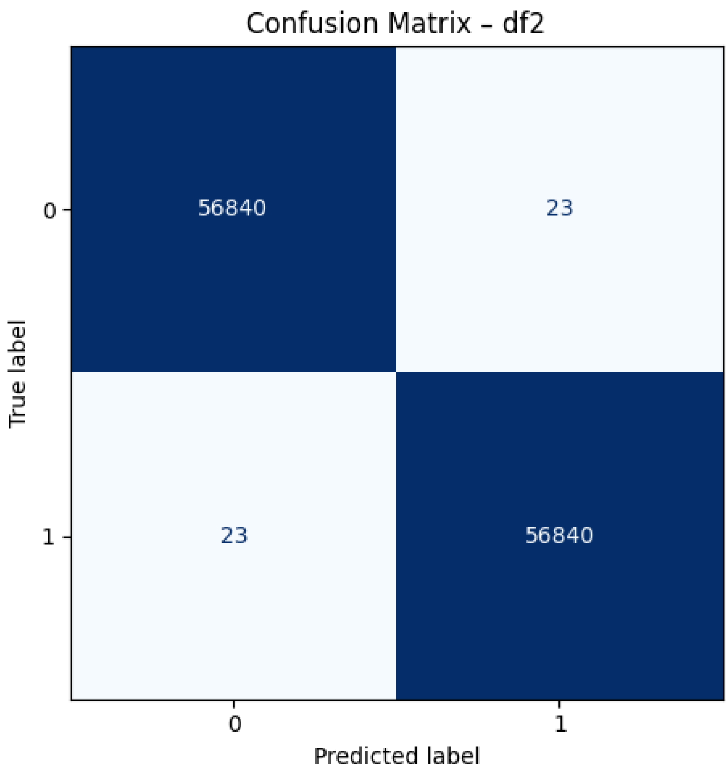
Testing Hybridized Mayfly and Pelican Algorithm on Second Dataset

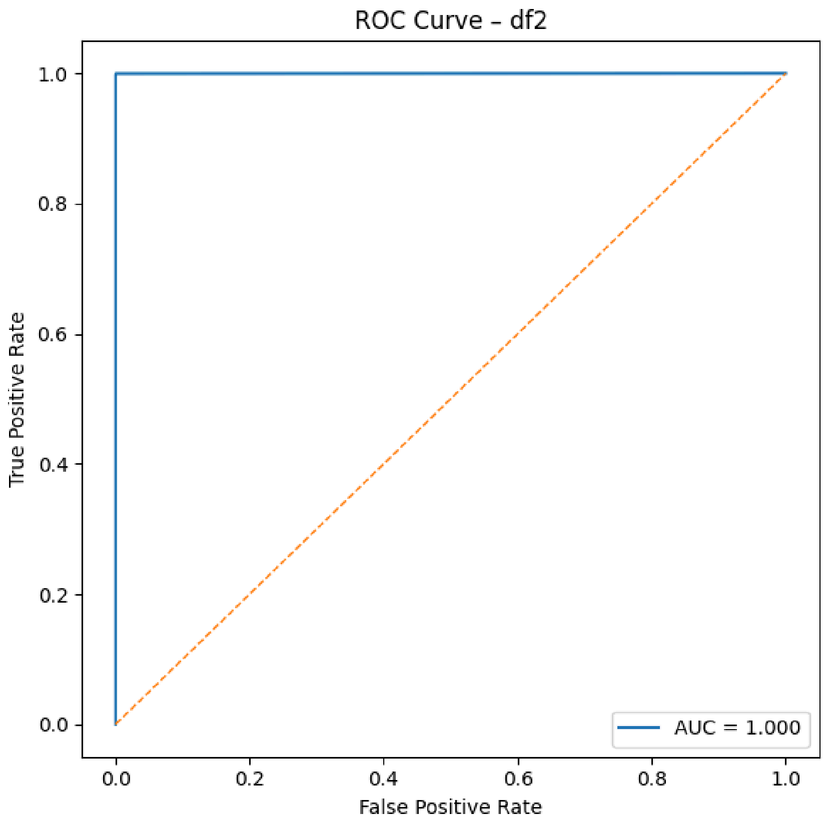
Evaluation on df2

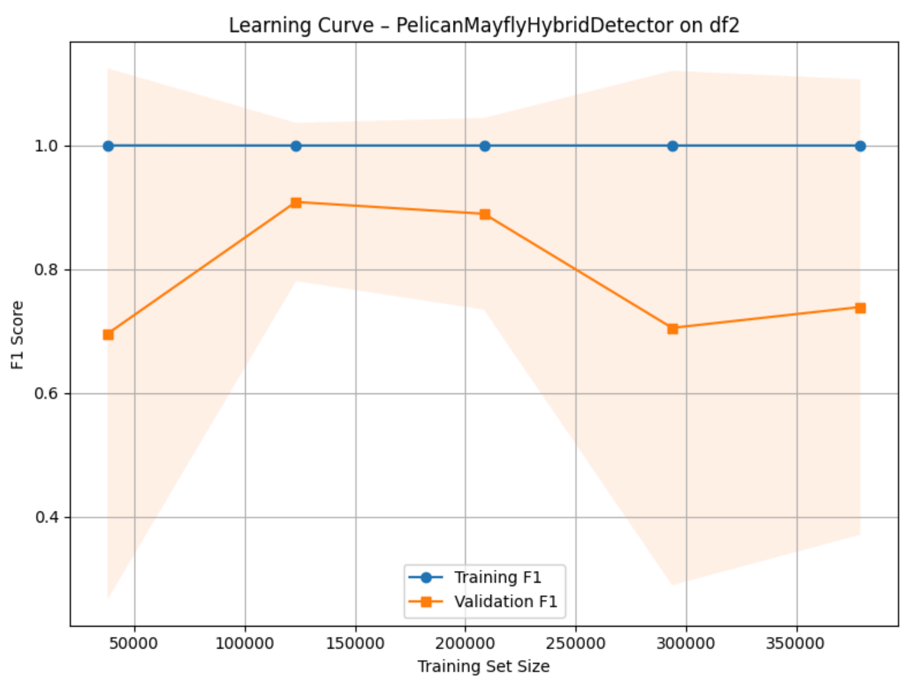
Performance Metrics

Metric	Value
Accuracy	0.9996
Precision	0.9996
Recall	0.9996
F1 Score	0.9996

ROC AUC	0.9998
R^2 (proba)	0.9986




ROC Curve – df2
True Positive Rate
False Positive Rate
AUC = 1.000
0.0
0.2
0.4
0.6
0.8
1.0

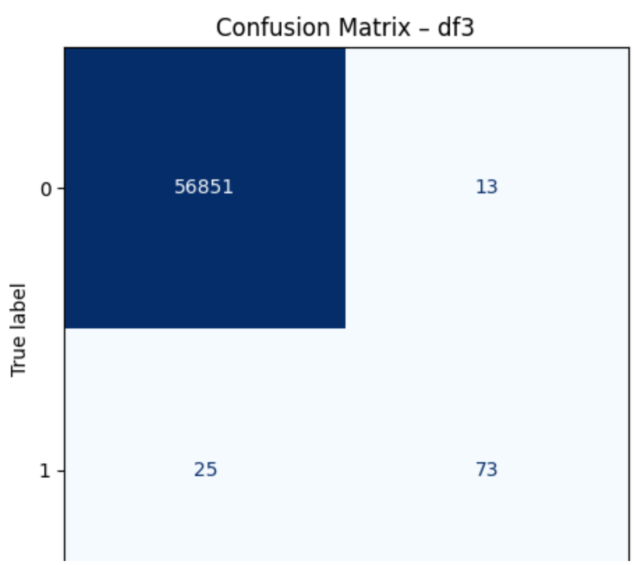

Learning Curve – PelicanMayflyHybridDetector on df2
F1 Score
Training Set Size
Training F1
Validation F1
0.4
0.6
0.8
1.0
50000
100000
150000
200000
250000
300000
350000

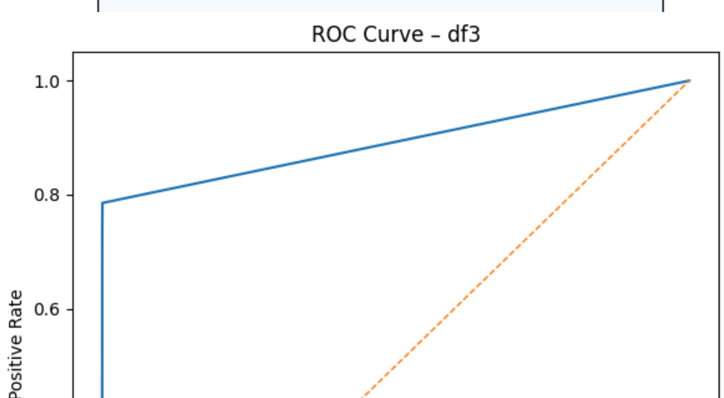
Testing Hybridized Mayfly and Pelican Algorithm on Third Dataset

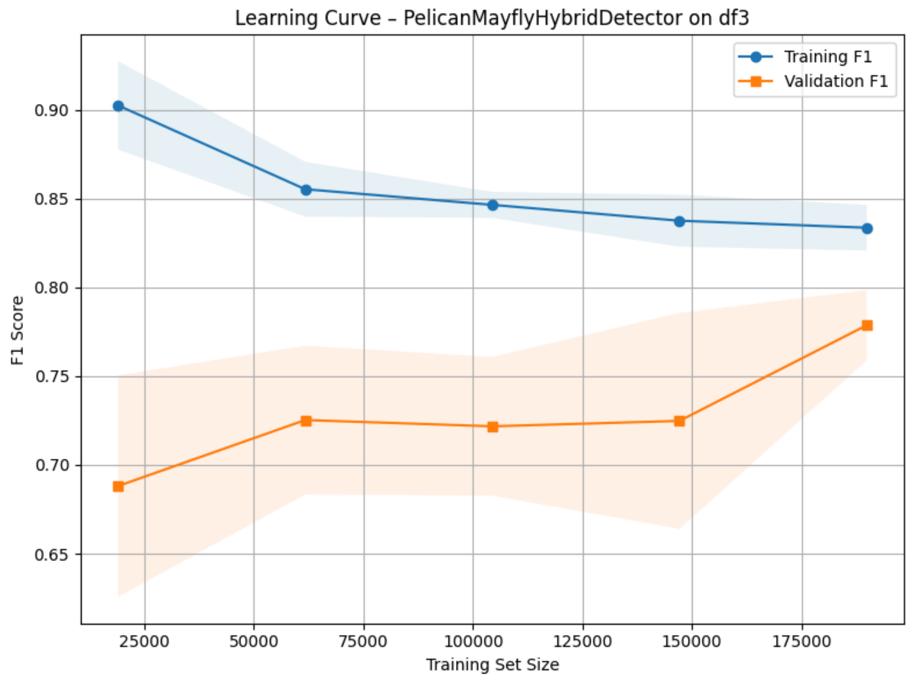
Evaluation on df3

Performance Metrics

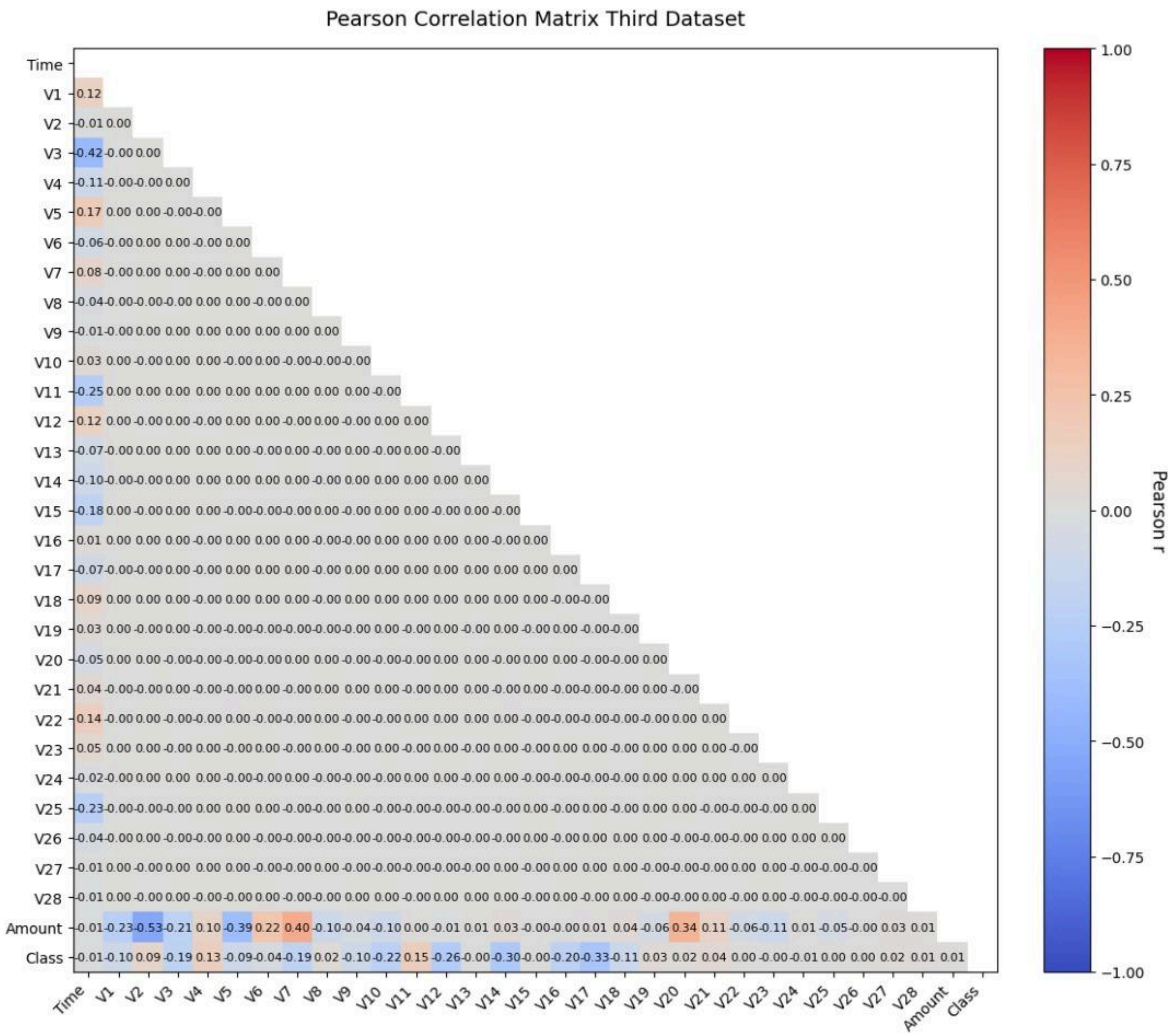
Metric	Value
Accuracy	0.9993
Precision	0.8488
Recall	0.7449
F1 Score	0.7935
ROC AUC	0.8926
R^2 (proba)	0.6342

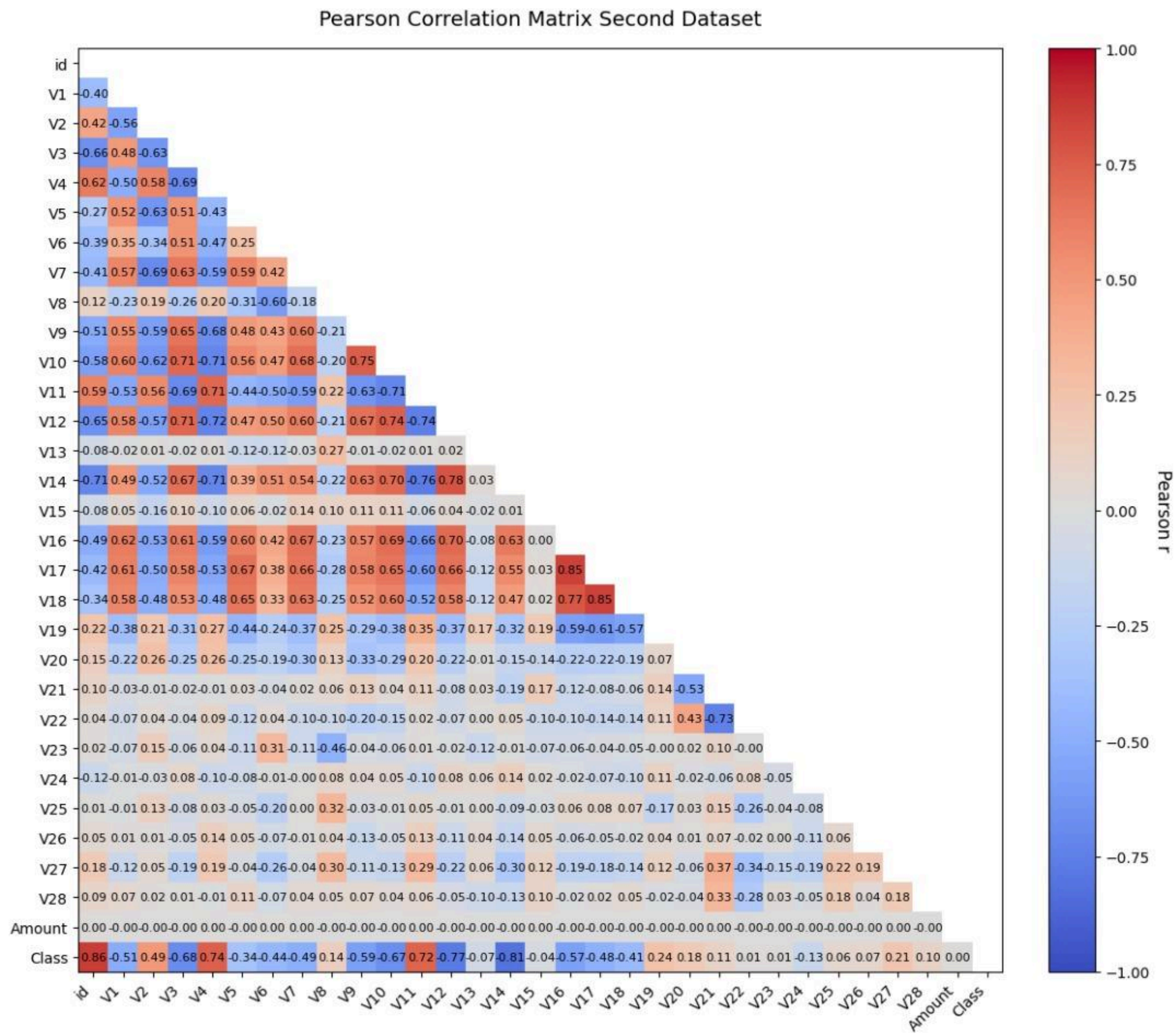




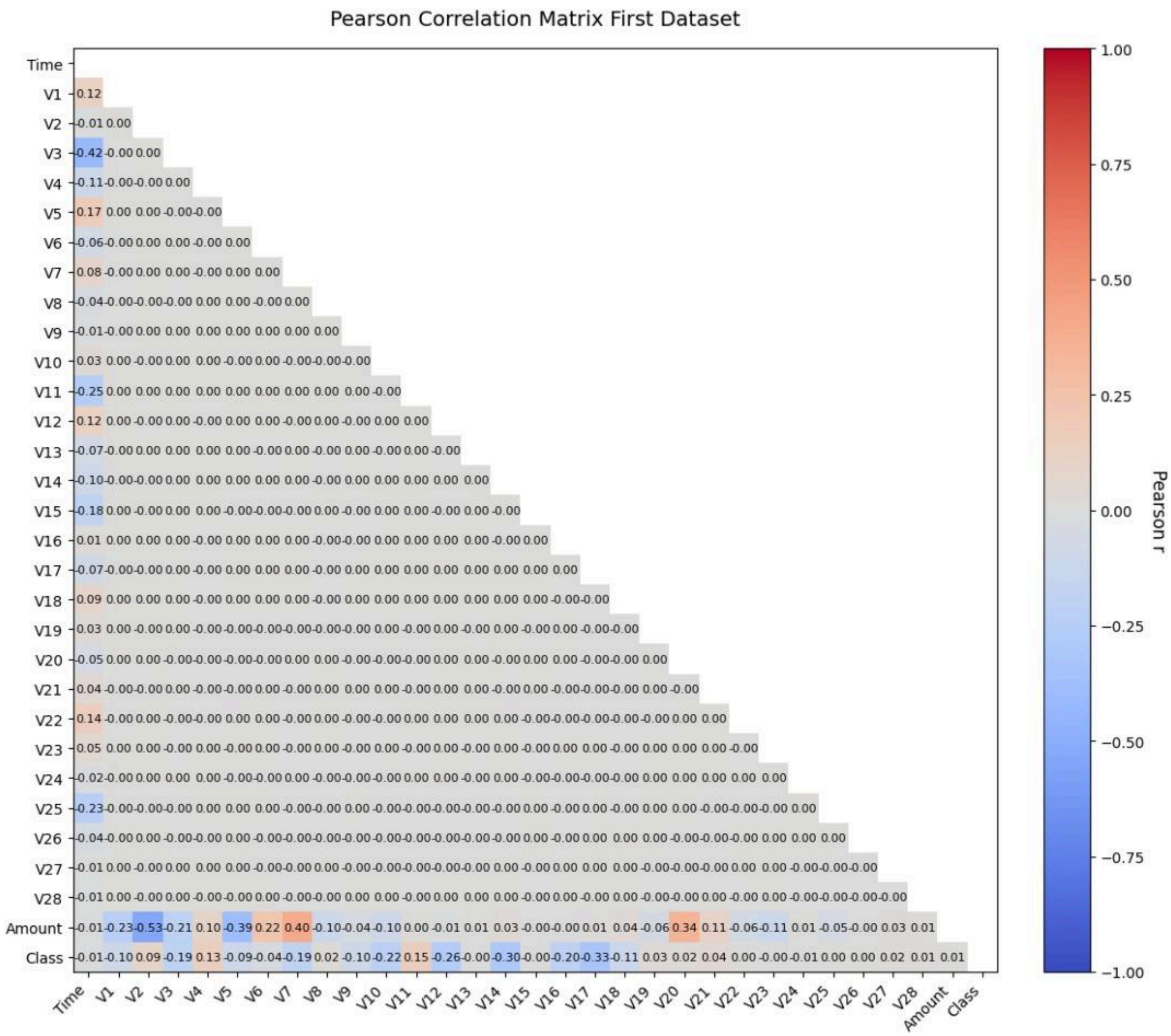


Learning Curve – PelicanMayflyHybridDetector on df3
Training F1
Validation F1
0.90
0.85
0.80
0.75
0.70
0.65
F1 Score
25000
50000
75000
100000
125000
150000
175000
Training Set Size

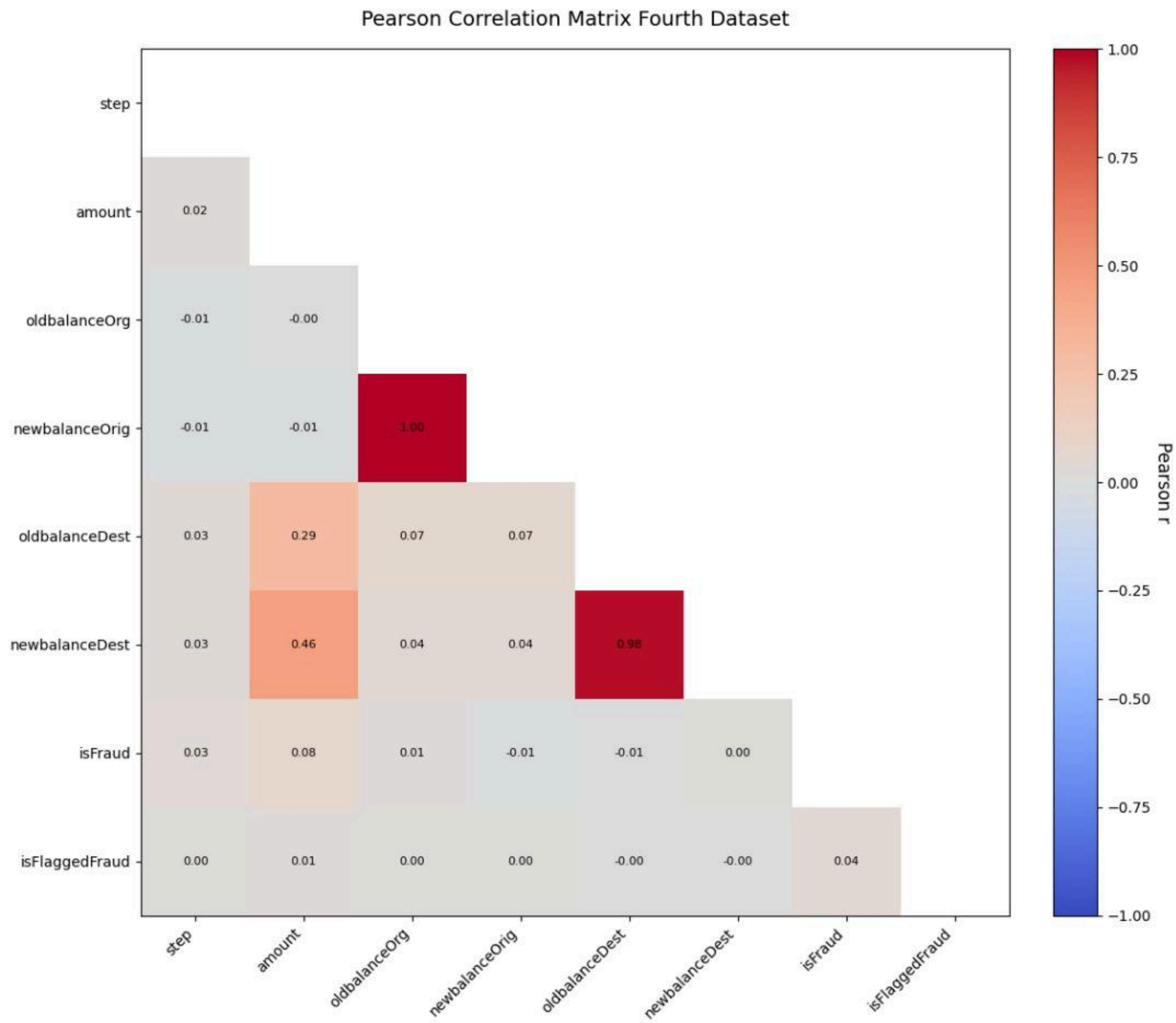

Pearson Correlation Matrix Third Dataset
Pearson r
Time
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12
V13
V14
V15
V16
V17
V18
V19
V20
V21
V22
V23
V24
V25
V26
V27
V28
Amount
Class



Pearson Correlation Matrix Second Dataset
Pearson r



Pearson Correlation Matrix First Dataset
Pearson r



Pearson Correlation Matrix Fourth Dataset
step
amount
oldbalanceOrg
newbalanceOrig
oldbalanceDest
newbalanceDest
isFraud
isFlaggedFraud
0.02
-0.01
-0.00
-0.01
-0.01
1.00
0.03
0.29
0.07
0.07
0.03
0.46
0.04
0.04
0.98
0.03
0.08
0.01
-0.01
-0.01
0.00
0.00
0.01
0.00
0.00
-0.00
-0.00
0.04
Pearson r
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Mayfly Algorithm Results

Classification Report and Test Results

Classification Report by Class

Class	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	56,864
1	0.74	0.86	0.80	98

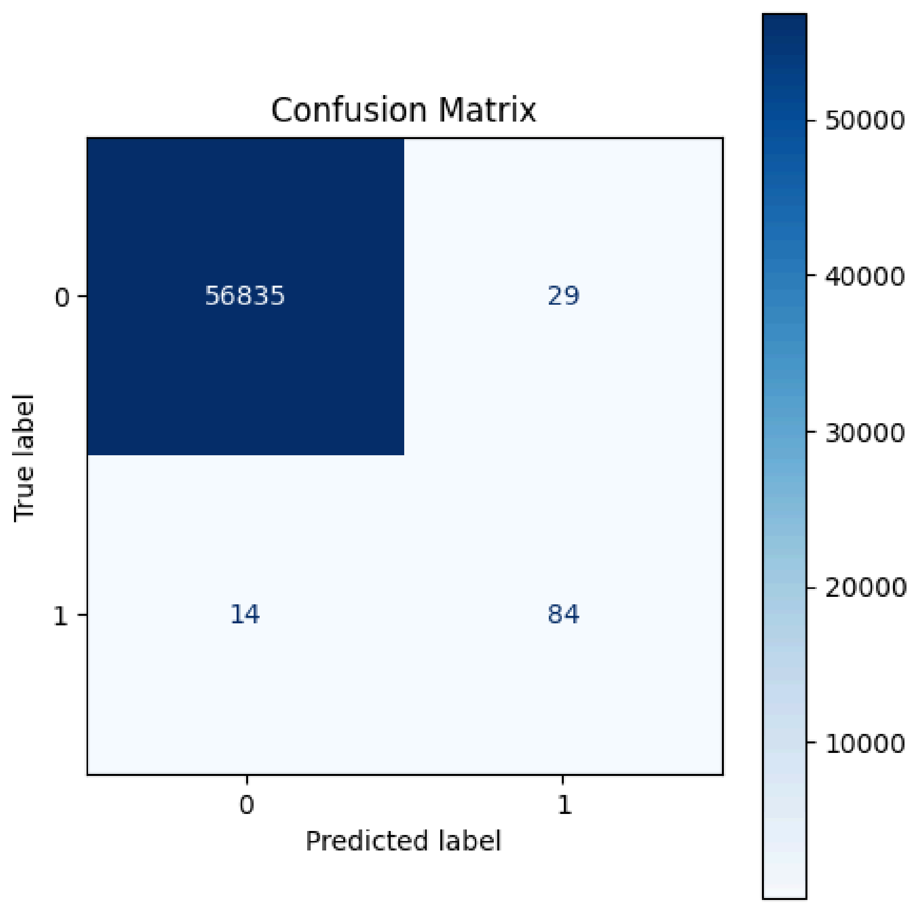
Summary Statistics

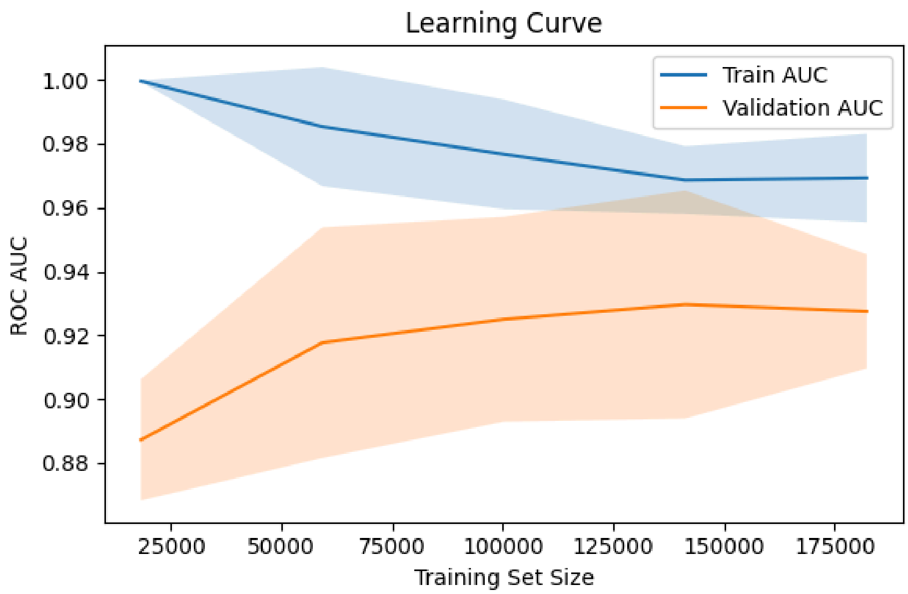
Metric Type	Precision	Recall	F1-Score	Support
Accuracy	-	-	1.00	56,962
Macro Avg	0.87	0.93	0.90	56,962
Weighted Avg	1.00	1.00	1.00	56,962

Test Set Performance Metrics

Metric	Value

Accuracy	0.9992
Precision	0.7434
Recall	0.8571
F1 Score	0.7962
ROC AUC	0.9419
R^2 (proba)	0.6390





Values After Testing With Second Data Set

Test Set Results

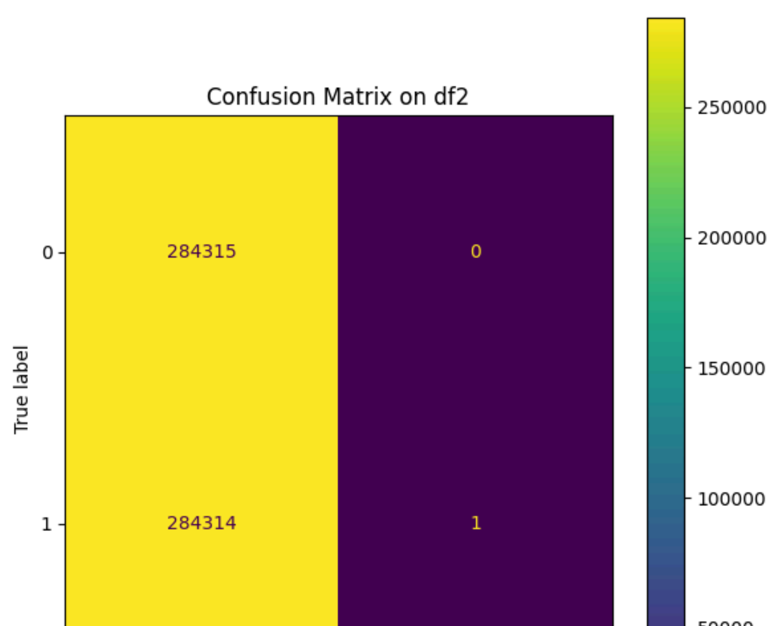
Performance Metrics

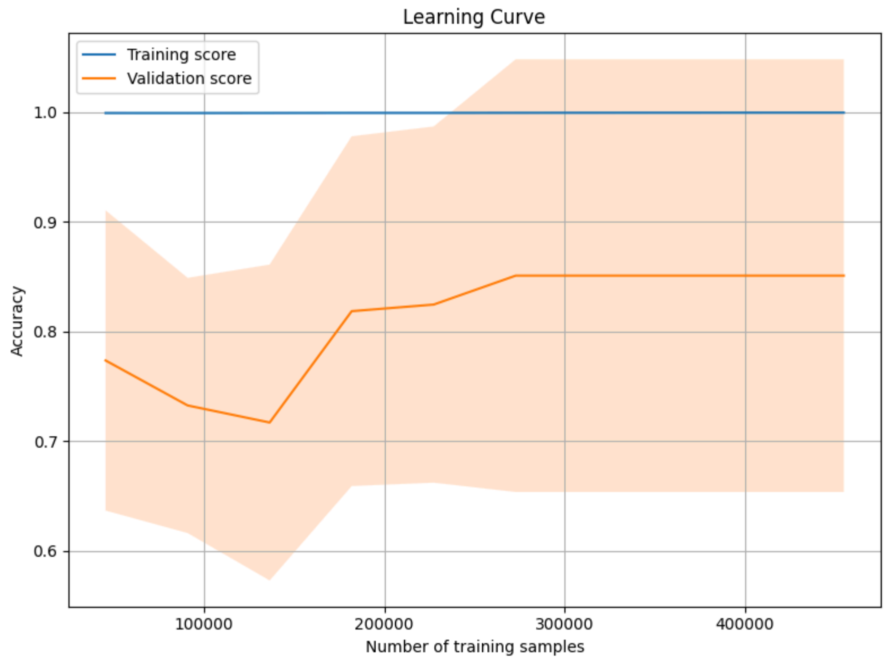
Metric	Value
Accuracy	0.5000
Precision	1.0000
Recall	0.0000
F1 Score	0.0000
ROC AUC	0.5149
R^2	-0.9981

Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	284,315	0
Actual Positive	284,314	1

- Total Samples: 568,630
- True Negatives: 284,315
- True Positives: 1
- False Negatives: 284,314
- False Positives: 0





Results After Tunning The Parameters with SIGMOID

Test Set Results Comparison

Calibrated + Threshold-tuned Results

Performance Metrics

Metric	Value
Accuracy	0.9997
Precision	0.9998
Recall	0.9997
F1 Score	0.9997
ROC AUC	0.9999

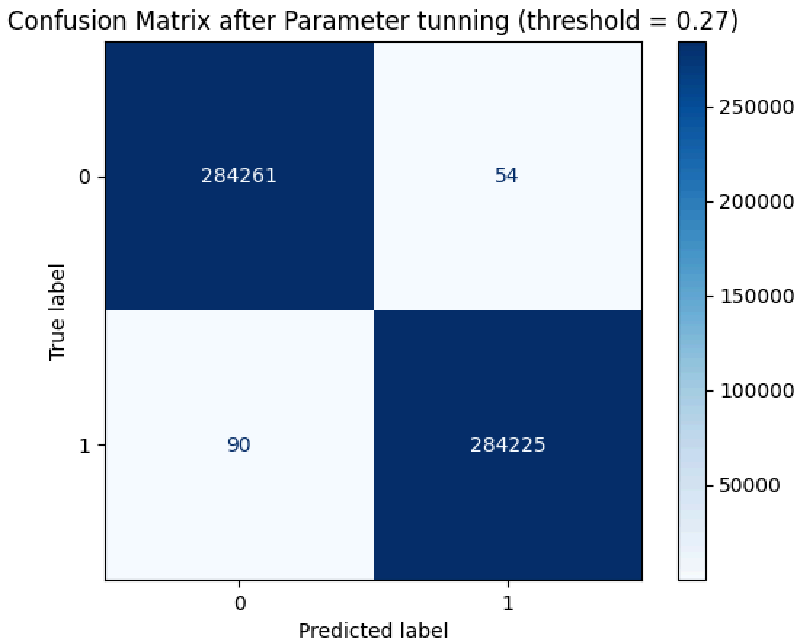
R^2 (proba)	0.9523

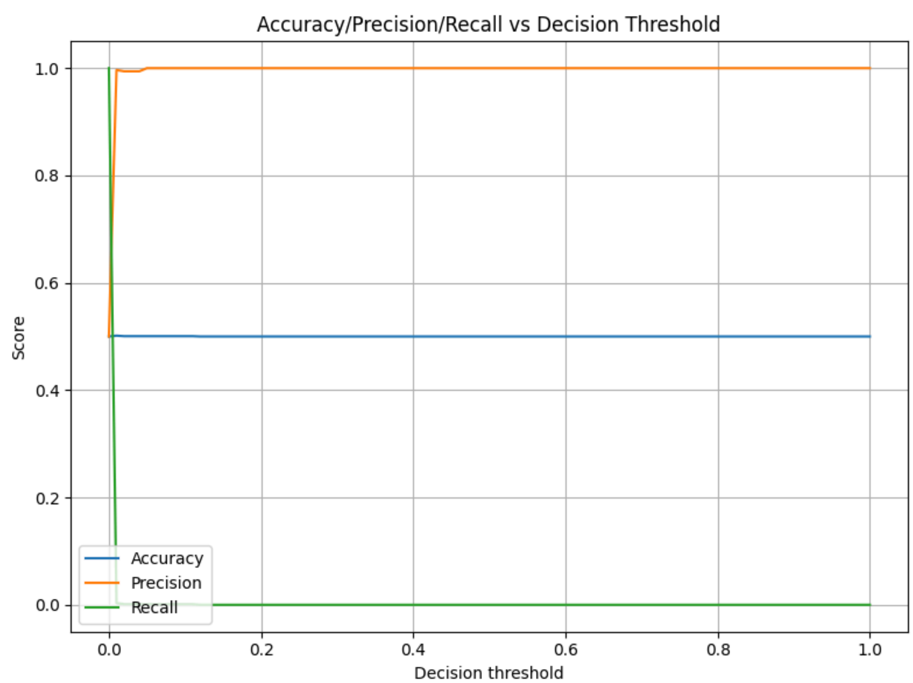
Confusion Matrix (Threshold: 0.27)

	Predicted Negative	Predicted Positive
Actual Negative	284,261	54
Actual Positive	90	284,225

Summary

- Total Samples: 568,630
- True Negatives: 284,261
- True Positives: 284,225
- False Negatives: 90
- False Positives: 54
- Optimal Threshold: 0.27


Confusion Matrix after Parameter tunning (threshold = 0.27)
284261
54
90
284225
True label
Predicted label
0
1
250000
200000
150000
100000
50000


Accuracy/Precision/Recall vs Decision Threshold
Score
Decision threshold
Accuracy
Precision
Recall

Values After Testing With Third Data Set

Test Set Results

Performance Metrics

Metric	Value
Accuracy	0.9992
Precision	0.7778
Recall	0.7256
F1 Score	0.7508
ROC AUC	0.8670
R^2	0.5429

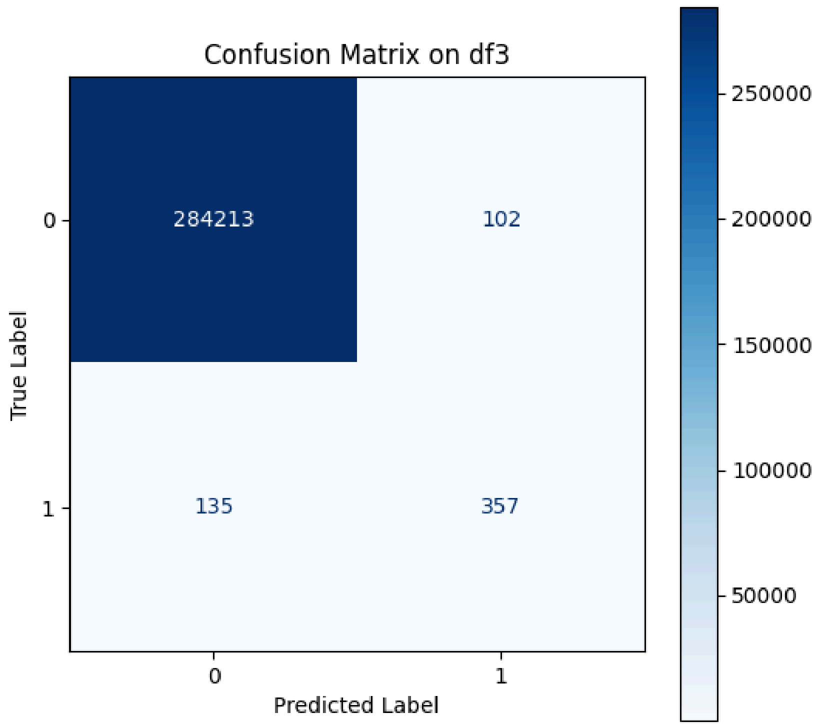
Confusion Matrix

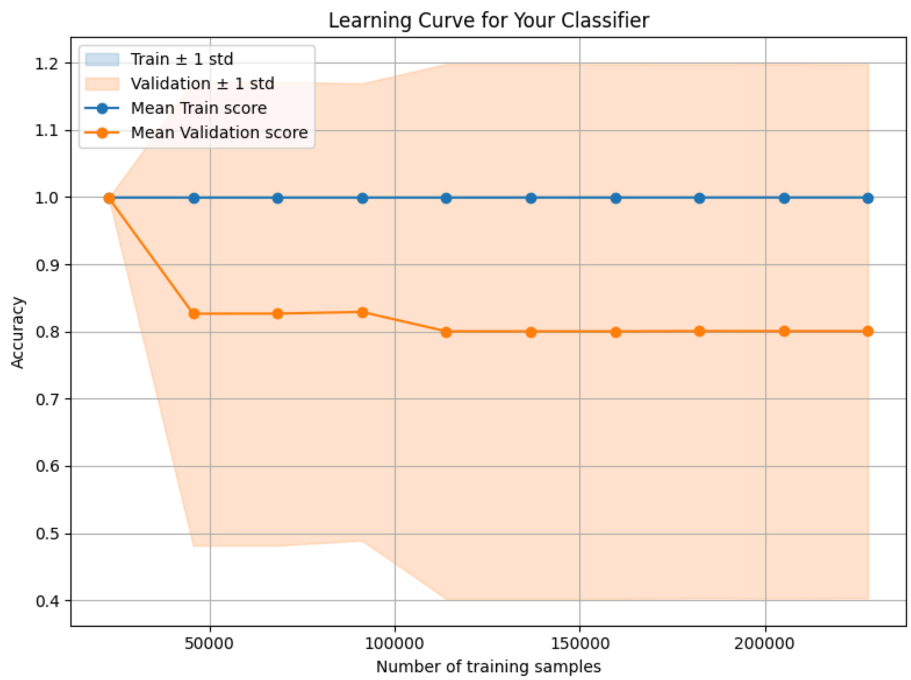
	Predicted Negative	Predicted Positive
Actual Negative	284,213	102
Actual Positive	135	357

Summary

- Total Samples: 284,807
- True Negatives: 284,213

- True Positives: 357
- False Negatives: 135
- False Positives: 102



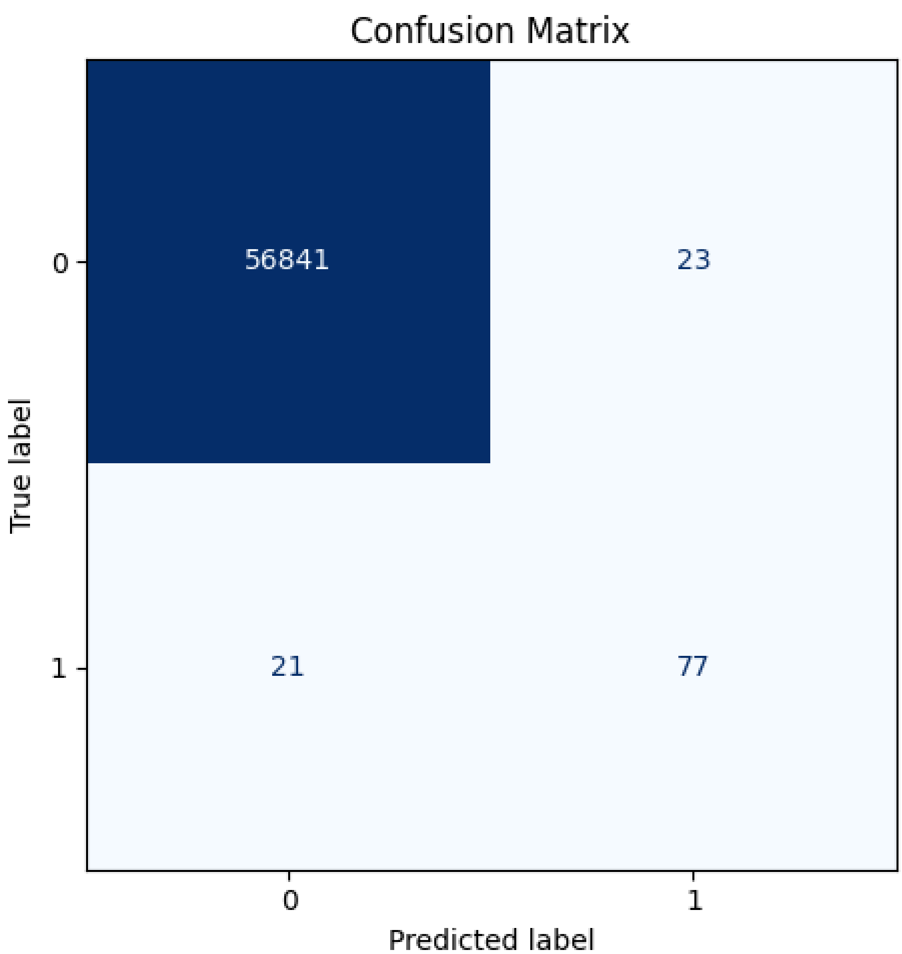


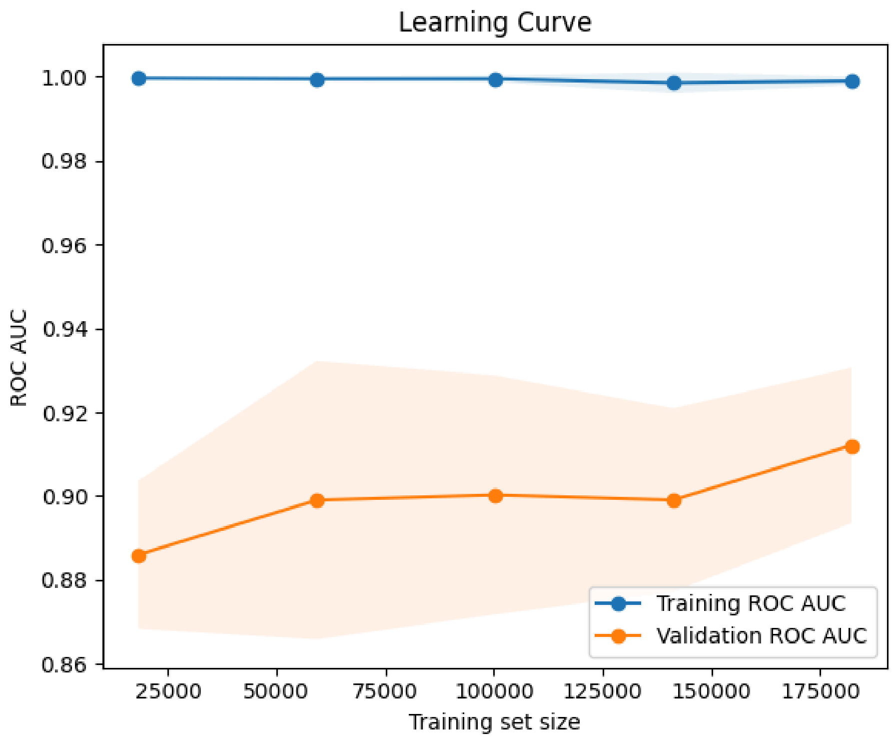
Pelican Model Results

Test Set Results

Metric	Value

Accuracy	0.9992
Precision	0.7700
Recall	0.7857
F1 Score	0.7778
ROC AUC	0.9244
R^2 (proba)	0.5963





Values From the Second Test Dataset

Test Set Results Comparison

Calibrated Results on New Data

Performance Metrics

Metric	Value
Accuracy	0.9997
Precision	0.9997
Recall	0.9996
F1 Score	0.9997
ROC AUC	0.9999

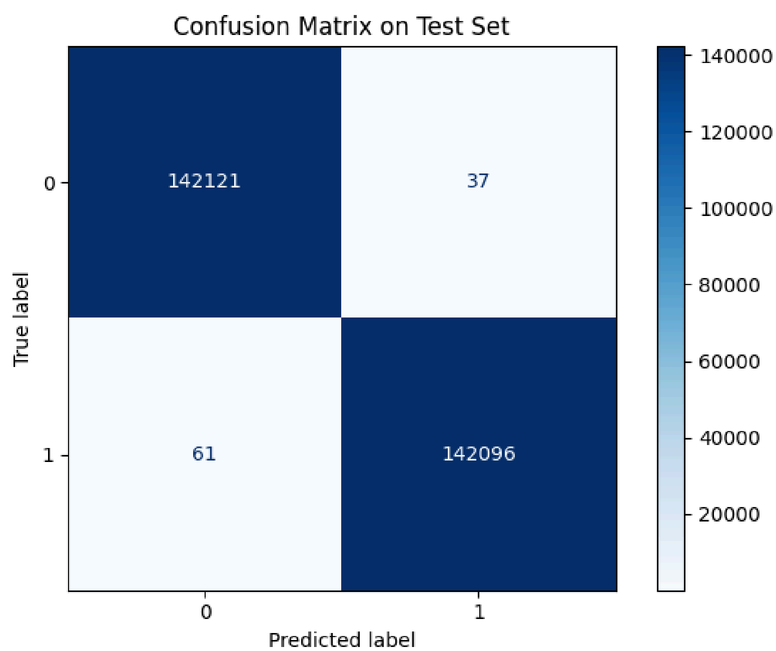
R^2 (proba)	0.9987

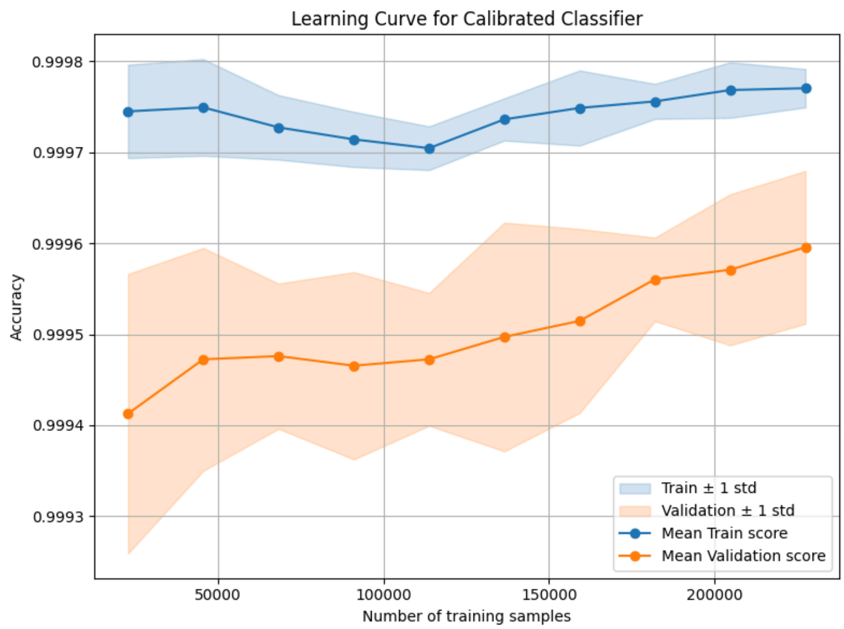
Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	142,121	37
Actual Positive	61	142,096

Summary

- **Total Samples**: 284,315
- **True Negatives**: 142,121
- **True Positives**: 142,096
- **False Negatives**: 61
- **False Positives**: 37





Values From the Third Test Dataset

Performance Metrics

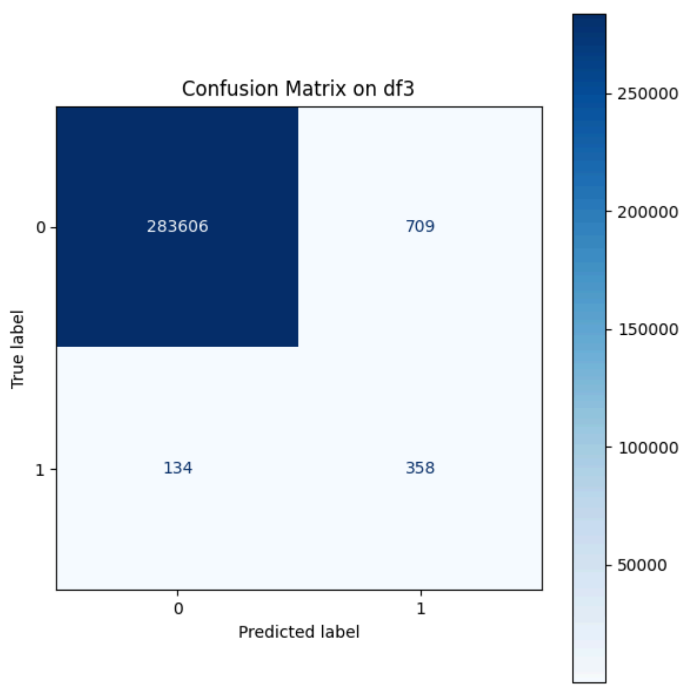
Metric	Value
Accuracy	0.9970
Precision	0.3355
Recall	0.7276
F1 Score	0.4593
ROC AUC	0.8393
R^2 (proba)	-0.7387

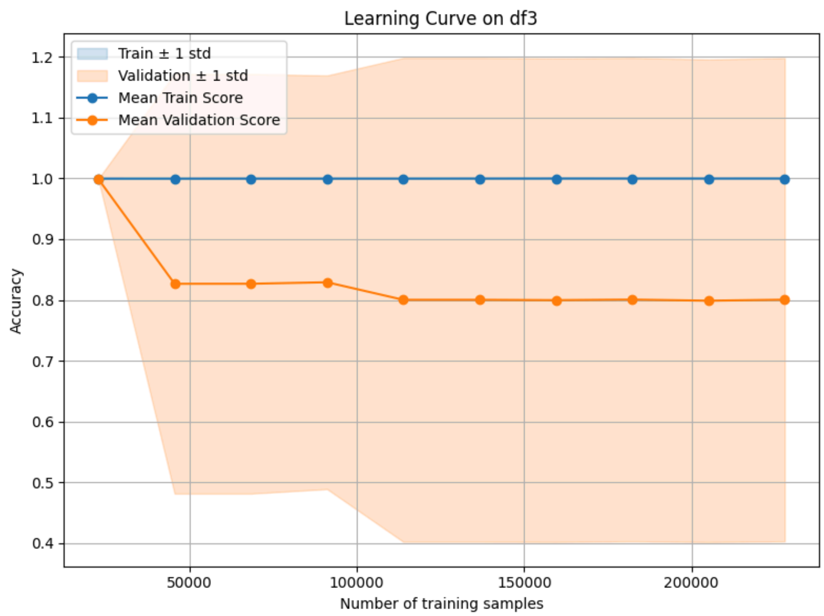
Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	283,606	709
Actual Positive	134	358

Summary

- **Total Samples**: 284,807
- **True Negatives**: 283,606
- **True Positives**: 358
- **False Negatives**: 134
- **False Positives**: 709




Learning Curve on df3
Train ± 1 std
Validation ± 1 std
Mean Train Score
Mean Validation Score
Accuracy
Number of training samples
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
50000
100000
150000
200000