# “Online Signature Verification using Machine Learning Algorithms”

RAJAT SAINI (S0I2UA)

## 1. Introduction

Secure identity verification remains a core challenge in the digital era. Signatures are still widely used in sectors like banking, legal, and administration, but manual verification is often unreliable and vulnerable to forgery. This project aims to automate the process using machine learning to enhance accuracy, reduce human error, and improve trust in signature-based systems.

### Motivation

Even with modern biometrics, handwritten signatures retain legal importance. Forged signatures can lead to fraud and legal issues. A machine learning approach offers a scalable and objective way to improve security and efficiency in systems relying on signature verification.

---

## 2. Introduction to Algorithms

This project used three supervised machine learning models—**Random Forest**, **KNN**, and **SVM**—chosen for their strengths in classification and suitability for biometric tasks like signature verification.

### 2.1 Random Forest

Random Forest builds multiple decision trees and predicts based on majority voting, offering strong performance on both linear and non-linear data with good resistance to overfitting.

- **Pros**: High accuracy, robust to overfitting, handles high-dimensional data.
- **Cons**: Slower for real-time use, less interpretable.

### 2.2 K-Nearest Neighbors (KNN)

KNN classifies samples based on the labels of their nearest neighbors and works well for smaller datasets due to its simplicity and non-parametric nature.

- **Pros**: Simple, effective for non-linear patterns, no assumptions about data.
- **Cons**: Slow during inference, sensitive to dimensionality.

### 2.3 Support Vector Machine (SVM)

SVM finds an optimal separating hyperplane; with an RBF kernel, it effectively captures non-linear patterns and performs well in high-dimensional spaces.

- **Pros**: Accurate, generalizes well, suitable for complex data.
- **Cons**: Sensitive to parameter settings, slower on large datasets.

---

## 3. Datasets Overview

Five datasets provided by my professor were used for training and testing, each containing genuine and forged signatures with varying features and structures.

### ➢ MCYT Dataset

- **Samples**: 20000 train / 1000 test
- **Features**: X, Y, P, al, az
- Largest dataset; uses all five features. Data was aggregated per SigID by computing feature means.

### ➢ SVC Dataset

- **Samples**: 5120 train / 320 test
- **Features**: X, Y, P, al, az
- **Notes**: Fully structured and clean; used directly after label mapping and scaling.

### ➢ Chinese Dataset

- **Samples**: 4124 train / 285 test
- **Features**: X, Y, P
- **Notes**: Compact and clean dataset. Orientation features were unavailable.

### ➢ Dutch Dataset

- **Samples**: 7056 train / 502 test
- **Features**: X, Y, P
- **Notes**: No orientation data. Emphasizes pressure and spatial patterns.

### ➢ German Dataset

- **Samples**: 1778 train / 151 test
- **Features**: X, Y, P
- **Notes**: Smallest test set; used to test robustness with fewer features.

### ➢ Common Preprocessing Steps

- signatureOrigin mapped to binary labels (1 = Genuine, 0 = Forged)
- Columns stripped of whitespace
- MinMaxScaler applied for normalization
- Datasets split into clean training and testing CSV files

## 4. Methodology

This project developed and assessed machine learning models for online signature verification through preprocessing, training, evaluation, and visualization.

### 4.1 Preprocessing

All datasets (provided in CSV format with training and testing files) underwent the following steps:

- **Label Mapping:** A Label column was added by mapping "Genuine" to 1 and "Forged" to 0.
- **Feature Selection:** Valid features (X, Y, P, and when available al, az) were retained by removing any with missing values.
- **MCYT Aggregation:** Since MCYT data included multiple points per signature, it was aggregated by averaging feature values per SigID.
- **Normalization:** All features were scaled to the [0, 1] range using MinMaxScaler to ensure model stability and performance.

### 4.2 Model Training & Evaluation

Three classification models were applied to each dataset:

- **Random Forest (RF):** Ensemble of decision trees, effective with high-dimensional, tabular data.
- **K-Nearest Neighbors (KNN):** Distance-based model that classifies based on nearest training samples.
- **Support Vector Machine (SVM):** Kernel-based method optimized for high-dimensional classification.

Each model was trained on the scaled training data and evaluated on the test set in a uniform loop.

### 4.3 Evaluation Metrics

To assess performance, the following metrics were calculated from the confusion matrix:

- **Accuracy:** Correct predictions out of total samples.
- **Precision:** TP / (TP + FP), i.e., correct genuine predictions.
- **F1 Score:** 2 × (Precision × Recall) / (Precision + Recall), balancing precision and recall.
- **FAR:** Rate of incorrectly accepted forged signatures.
- **FRR:** Rate of incorrectly rejected genuine signatures.
- **EER:** Average of FAR and FRR, commonly used in biometric evaluations.

**4.4 Visualizations**: To interpret the results:

- **Confusion Matrix Heatmaps** were plotted per model and dataset.
- **PCA Scatter Plots** were used to visualize class separation by reducing feature space to 2D.

This ensured a consistent, repeatable evaluation pipeline across all signature datasets.

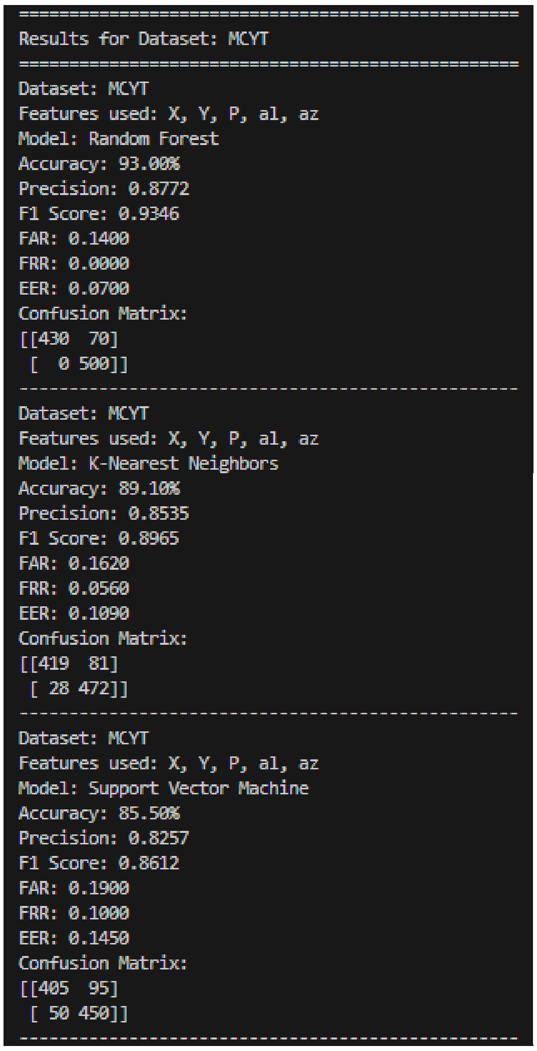
## 5. Results

To evaluate the performance of each machine learning algorithm on the five signature datasets, we calculated six key metrics: **Accuracy, Precision, F1 Score, FAR (False Acceptance Rate), FRR (False Rejection Rate), and EER (Equal Error Rate)**. Additionally, confusion matrices and PCA visualizations were generated for each dataset. The results are summarized below:

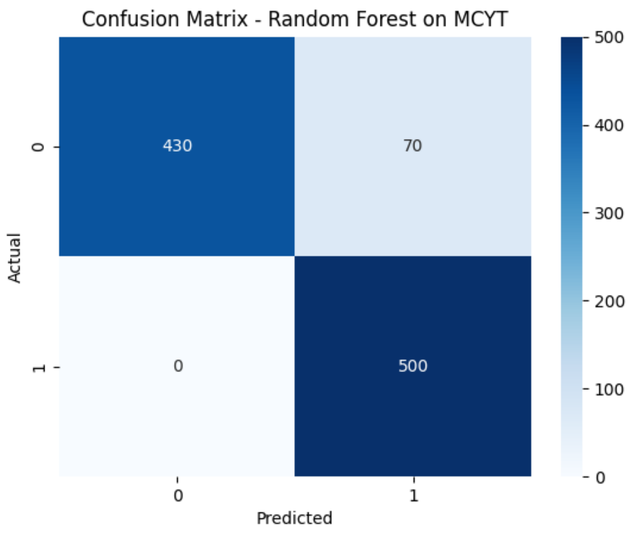
### 5.1 MCYT Dataset

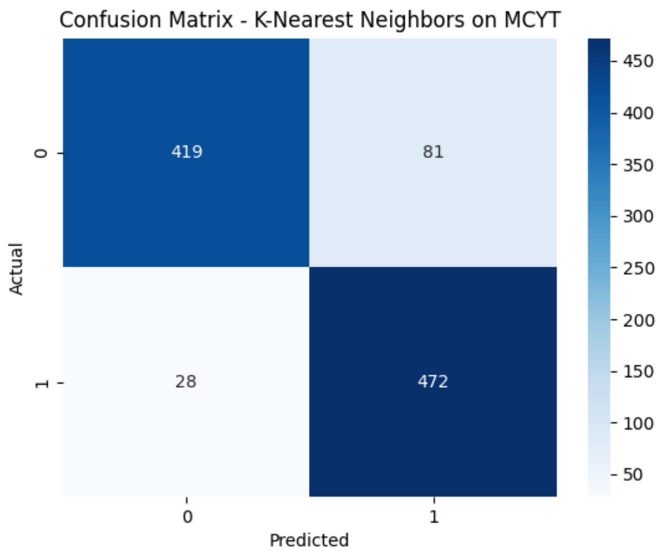
| **Model** | **Accuracy** | **Precision** | **F1 Score** | **FAR** | **FRR** | **EER** |
|---|---|---|---|---|---|---|
| Random Forest | 93.00% | 0.8772 | 0.9346 | 0.1400 | 0.0000 | 0.0700 |
| K-Nearest Neighbors | 89.10% | 0.8535 | 0.8965 | 0.1620 | 0.0560 | 0.1090 |
| Support Vector Machine | 85.50% | 0.8257 | 0.8612 | 0.1900 | 0.1000 | 0.1450 |

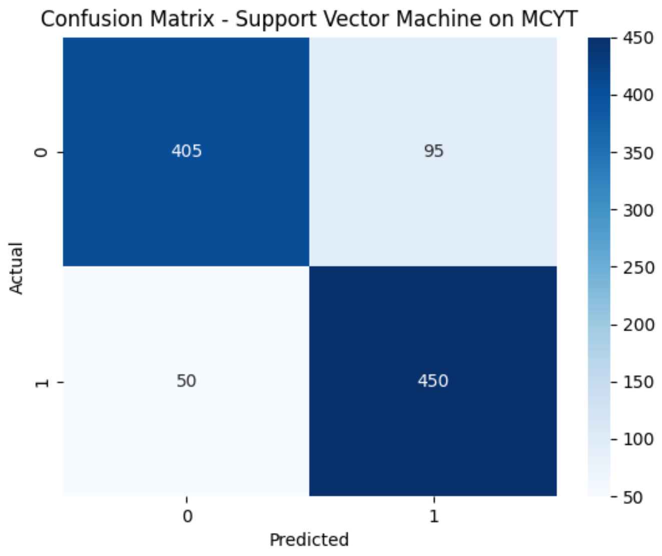
**Highlight**: Random Forest achieved the best performance across all metrics, especially with a perfect FRR of 0.0000, indicating no genuine signatures were wrongly rejected.

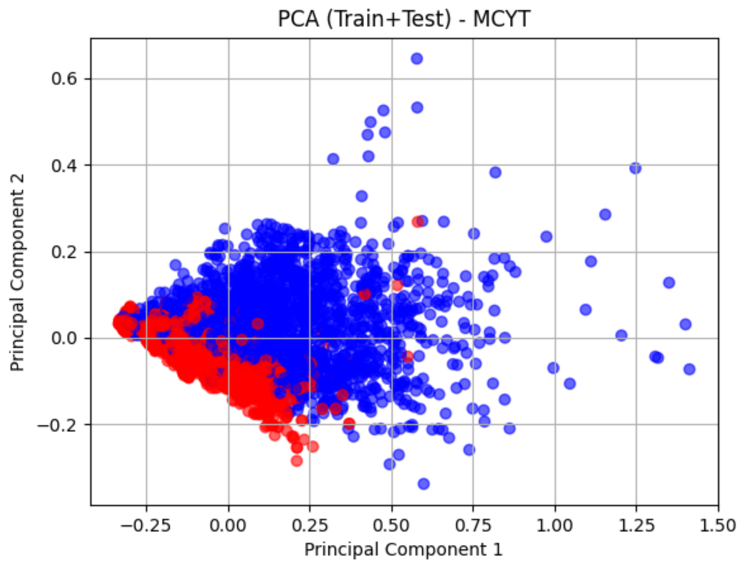


```
==================================================
Results for Dataset: MCYT
==================================================
Dataset: MCYT
Features used: X, Y, P, al, az
Model: Random Forest
Accuracy: 93.00%
Precision: 0.8772
F1 Score: 0.9346
FAR: 0.1400
FRR: 0.0000
EER: 0.0700
Confusion Matrix:
[[430  70]
 [  0 500]]
--------------------------------------------------
Dataset: MCYT
Features used: X, Y, P, al, az
Model: K-Nearest Neighbors
Accuracy: 89.10%
Precision: 0.8535
F1 Score: 0.8965
FAR: 0.1620
FRR: 0.0560
EER: 0.1090
Confusion Matrix:
[[419  81]
 [ 28 472]]
--------------------------------------------------
Dataset: MCYT
Features used: X, Y, P, al, az
Model: Support Vector Machine
Accuracy: 85.50%
Precision: 0.8257
F1 Score: 0.8612
FAR: 0.1900
FRR: 0.1000
EER: 0.1450
Confusion Matrix:
[[405  95]
 [ 50 450]]
--------------------------------------------------
```




Confusion Matrix - K-Nearest Neighbors on MCYT
419
81
28
472
Actual
Predicted
0
1
450
400
350
300
250
200
150
100
50

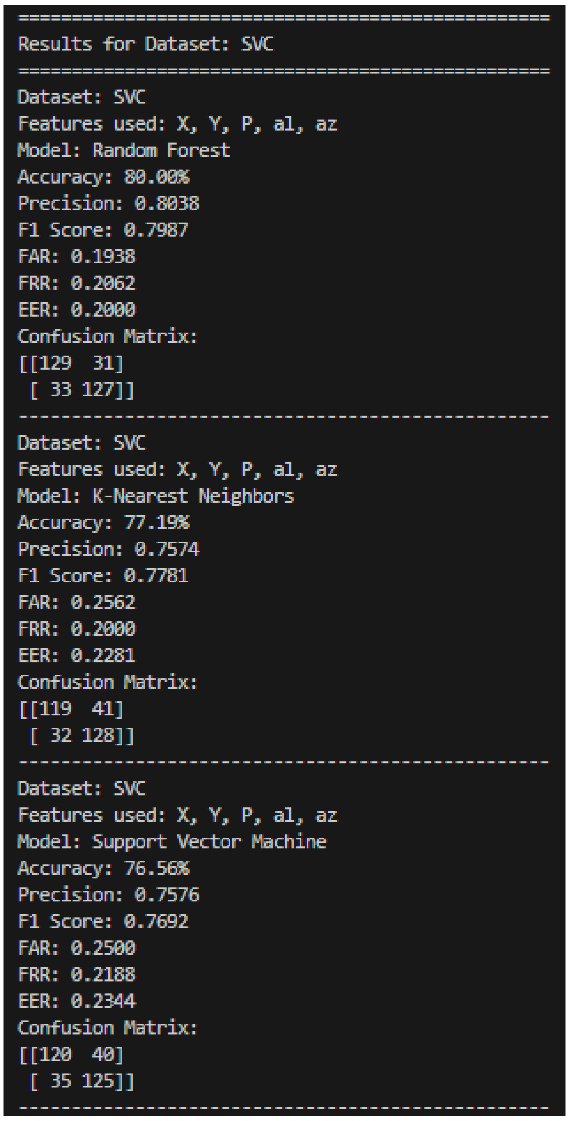

Confusion Matrix - Support Vector Machine on MCYT
405
95
50
450
Actual
Predicted
0
1
450
400
350
300
250
200
150
100
50


PCA (Train+Test) - MCYT
Principal Component 1
Principal Component 2
0.6
0.4
0.2
0.0
−0.2
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50

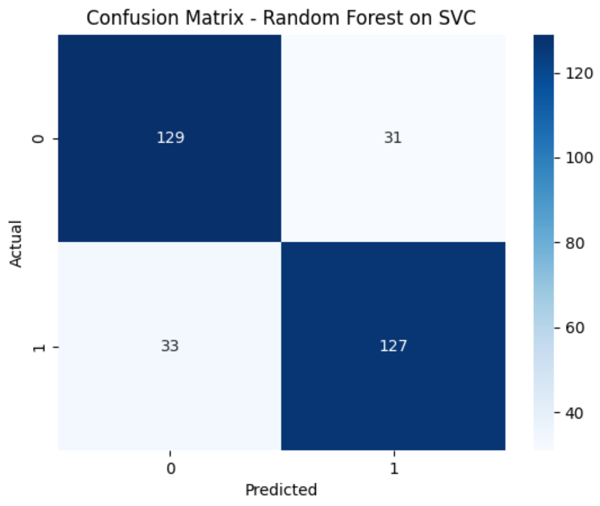
## 5.2 SVC Dataset

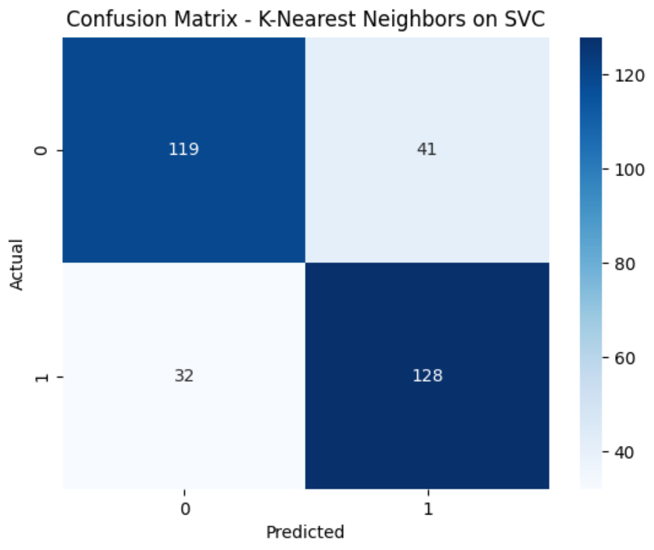
| Model | Accuracy | Precision | F1 Score | FAR | FRR | EER |
|---|---|---|---|---|---|---|
| Random Forest | 80.00% | 0.8038 | 0.7987 | 0.1938 | 0.2062 | 0.2000 |
| K-Nearest Neighbors | 77.19% | 0.7574 | 0.7781 | 0.2562 | 0.2000 | 0.2281 |
| Support Vector Machine | 76.56% | 0.7576 | 0.7692 | 0.2500 | 0.2188 | 0.2344 |

**Highlight**: Random Forest was again the top performer with balanced results, although all models struggled to lower both FAR and FRR simultaneously.

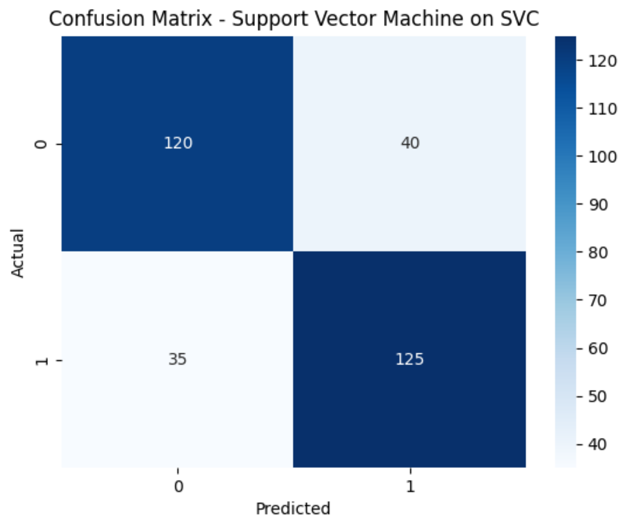


```
==================================================
Results for Dataset: SVC
==================================================
Dataset: SVC
Features used: X, Y, P, al, az
Model: Random Forest
Accuracy: 80.00%
Precision: 0.8038
F1 Score: 0.7987
FAR: 0.1938
FRR: 0.2062
EER: 0.2000
Confusion Matrix:
[[129  31]
 [ 33 127]]
--------------------------------------------------
Dataset: SVC
Features used: X, Y, P, al, az
Model: K-Nearest Neighbors
Accuracy: 77.19%
Precision: 0.7574
F1 Score: 0.7781
FAR: 0.2562
FRR: 0.2000
EER: 0.2281
Confusion Matrix:
[[119  41]
 [ 32 128]]
--------------------------------------------------
Dataset: SVC
Features used: X, Y, P, al, az
Model: Support Vector Machine
Accuracy: 76.56%
Precision: 0.7576
F1 Score: 0.7692
FAR: 0.2500
FRR: 0.2188
EER: 0.2344
Confusion Matrix:
[[120  40]
 [ 35 125]]
--------------------------------------------------
```

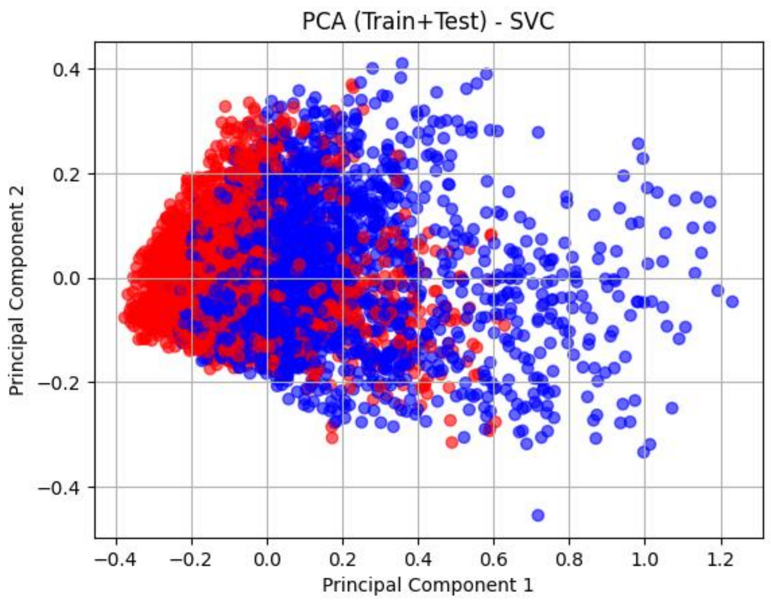




Confusion Matrix - K-Nearest Neighbors on SVC
Actual
Predicted
0
1
119
41
32
128
120
100
80
60
40



Confusion Matrix - Support Vector Machine on SVC
Actual
Predicted
0
1
120
40
35
125
120
110
100
90
80
70
60
50
40

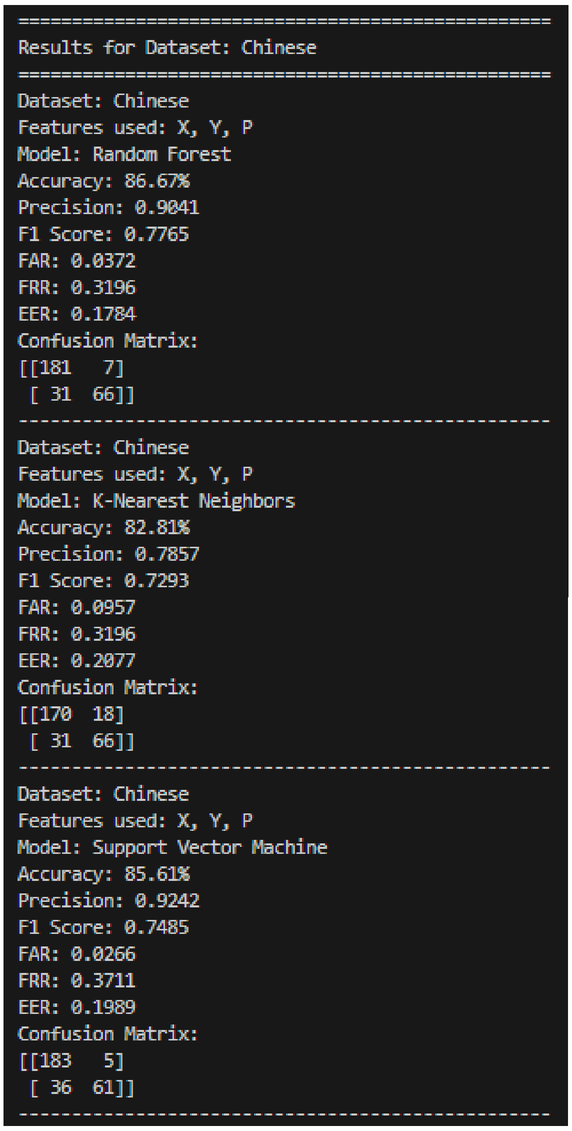


PCA (Train+Test) - SVC
Principal Component 1
Principal Component 2
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2

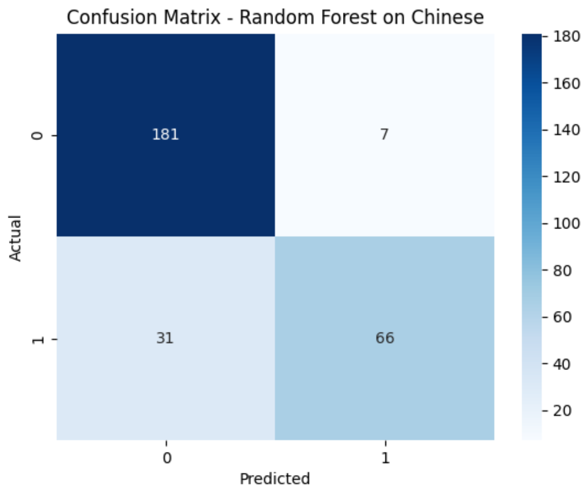
## 5.3 Chinese Dataset

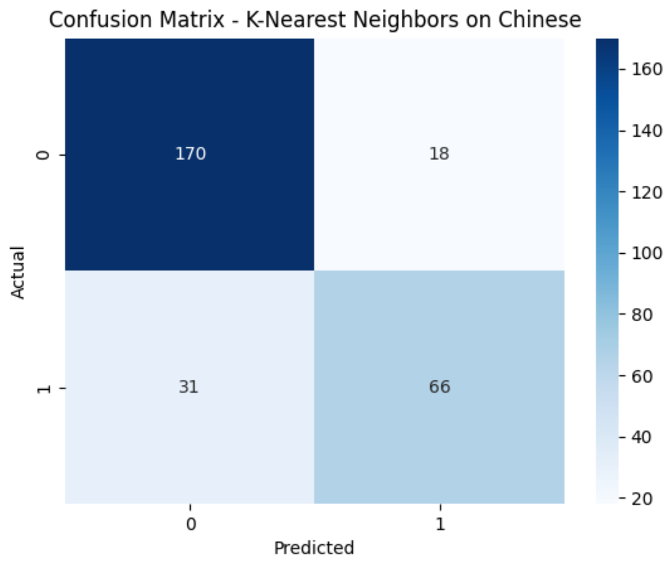
| Model | Accuracy | Precision | F1 Score | FAR | FRR | EER |
|---|---|---|---|---|---|---|
| Random Forest | 86.67% | 0.9041 | 0.7765 | 0.0372 | 0.3196 | 0.1784 |
| K-Nearest Neighbors | 82.81% | 0.7857 | 0.7293 | 0.0957 | 0.3196 | 0.2077 |
| Support Vector Machine | 85.61% | 0.9242 | 0.7485 | 0.0266 | 0.3711 | 0.1989 |

**Highlight**: Random Forest had the highest accuracy and well-balanced precision, though SVM achieved the best precision (0.9242) at the cost of high FRR.

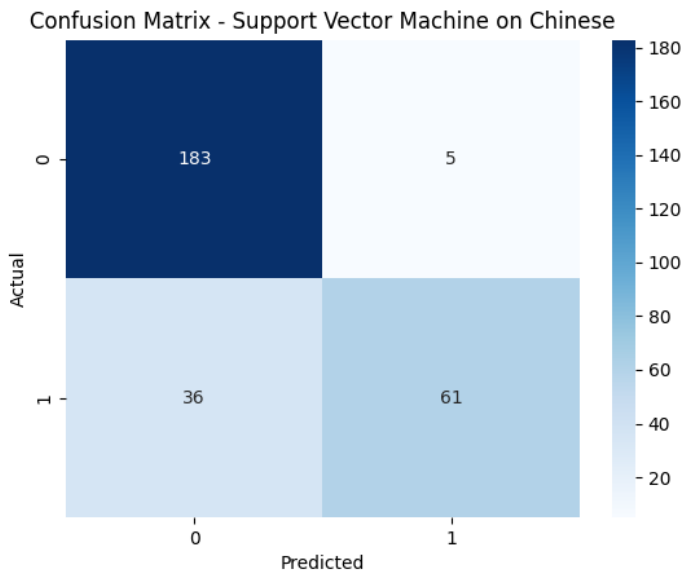


```
==================================================
Results for Dataset: Chinese
==================================================
Dataset: Chinese
Features used: X, Y, P
Model: Random Forest
Accuracy: 86.67%
Precision: 0.9041
F1 Score: 0.7765
FAR: 0.0372
FRR: 0.3196
EER: 0.1784
Confusion Matrix:
[[181   7]
 [ 31  66]]
--------------------------------------------------
Dataset: Chinese
Features used: X, Y, P
Model: K-Nearest Neighbors
Accuracy: 82.81%
Precision: 0.7857
F1 Score: 0.7293
FAR: 0.0957
FRR: 0.3196
EER: 0.2077
Confusion Matrix:
[[170  18]
 [ 31  66]]
--------------------------------------------------
Dataset: Chinese
Features used: X, Y, P
Model: Support Vector Machine
Accuracy: 85.61%
Precision: 0.9242
F1 Score: 0.7485
FAR: 0.0266
FRR: 0.3711
EER: 0.1989
Confusion Matrix:
[[183   5]
 [ 36  61]]
--------------------------------------------------
```

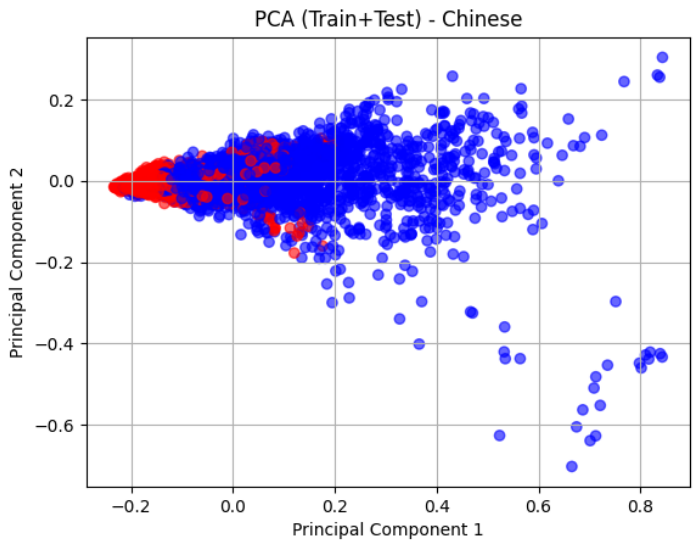




Confusion Matrix - K-Nearest Neighbors on Chinese
170
18
31
66
Actual
Predicted
0
1
160
140
120
100
80
60
40
20



Confusion Matrix - Support Vector Machine on Chinese
183
5
36
61
Actual
Predicted
0
1
180
160
140
120
100
80
60
40
20

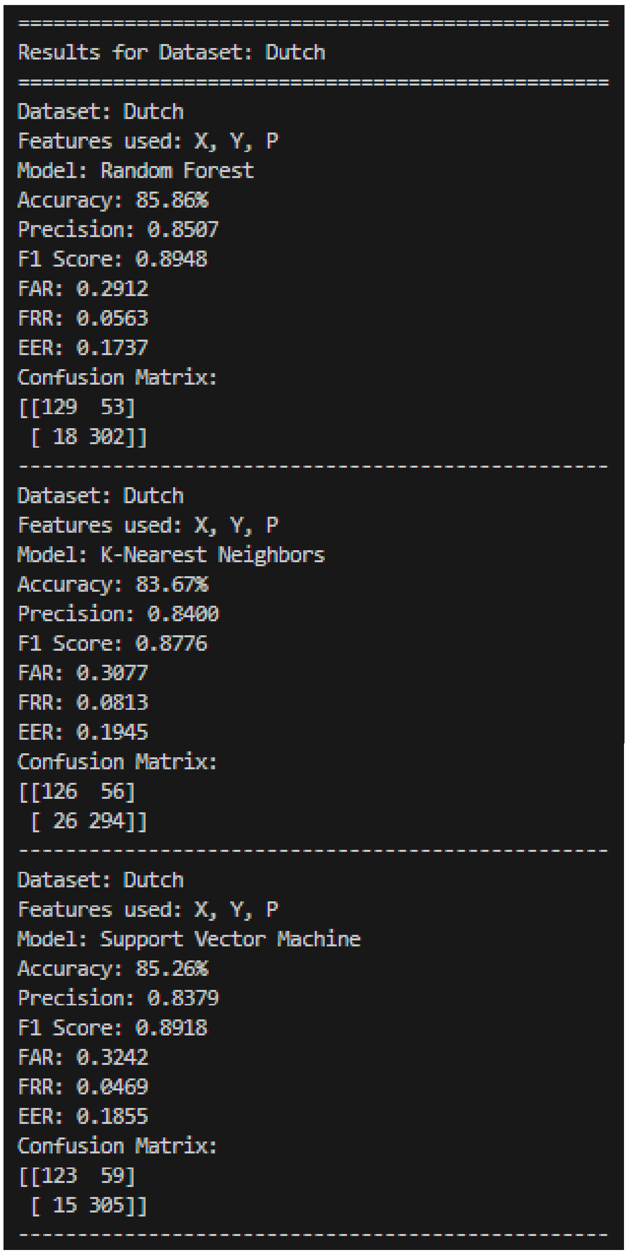


PCA (Train+Test) - Chinese
Principal Component 1
Principal Component 2
−0.2
0.0
0.2
0.4
0.6
0.8
0.2
0.0
−0.2
−0.4
−0.6

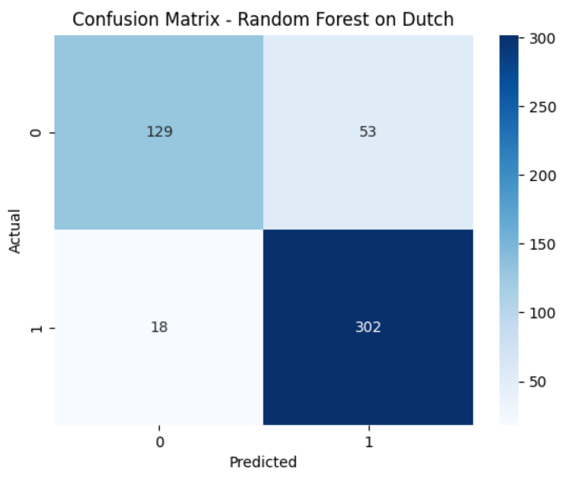
## 5.4 Dutch Dataset

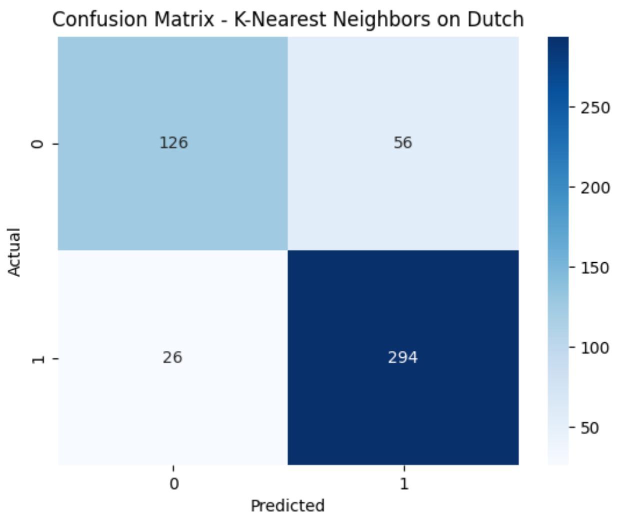
| **Model** | **Accuracy** | **Precision** | **F1 Score** | **FAR** | **FRR** | **EER** |
|---|---|---|---|---|---|---|
| Random Forest | 85.86% | 0.8507 | 0.8948 | 0.2912 | 0.0563 | 0.1737 |
| K-Nearest Neighbors | 83.67% | 0.8400 | 0.8776 | 0.3077 | 0.0813 | 0.1945 |
| Support Vector Machine | 85.26% | 0.8379 | 0.8918 | 0.3242 | 0.0469 | 0.1855 |

**Highlight**: Random Forest and SVM were closely matched, but Random Forest had better F1 and overall balance.

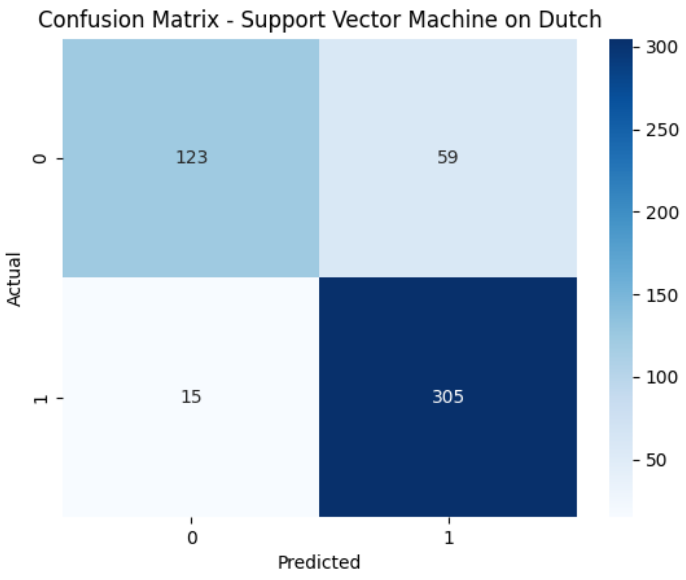


```
==================================================
Results for Dataset: Dutch
==================================================
Dataset: Dutch
Features used: X, Y, P
Model: Random Forest
Accuracy: 85.86%
Precision: 0.8507
F1 Score: 0.8948
FAR: 0.2912
FRR: 0.0563
EER: 0.1737
Confusion Matrix:
[[129  53]
 [ 18 302]]
--------------------------------------------------
Dataset: Dutch
Features used: X, Y, P
Model: K-Nearest Neighbors
Accuracy: 83.67%
Precision: 0.8400
F1 Score: 0.8776
FAR: 0.3077
FRR: 0.0813
EER: 0.1945
Confusion Matrix:
[[126  56]
 [ 26 294]]
--------------------------------------------------
Dataset: Dutch
Features used: X, Y, P
Model: Support Vector Machine
Accuracy: 85.26%
Precision: 0.8379
F1 Score: 0.8918
FAR: 0.3242
FRR: 0.0469
EER: 0.1855
Confusion Matrix:
[[123  59]
 [ 15 305]]
--------------------------------------------------
```

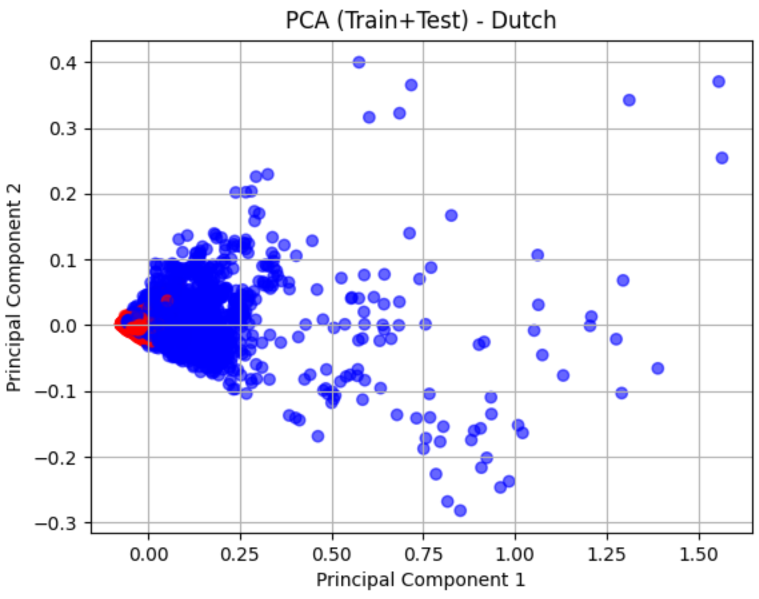




Confusion Matrix - K-Nearest Neighbors on Dutch
126
56
26
294
Actual
Predicted
0
1
0
1
250
200
150
100
50



Confusion Matrix - Support Vector Machine on Dutch
123
59
15
305
Actual
Predicted
0
1
0
1
300
250
200
150
100
50

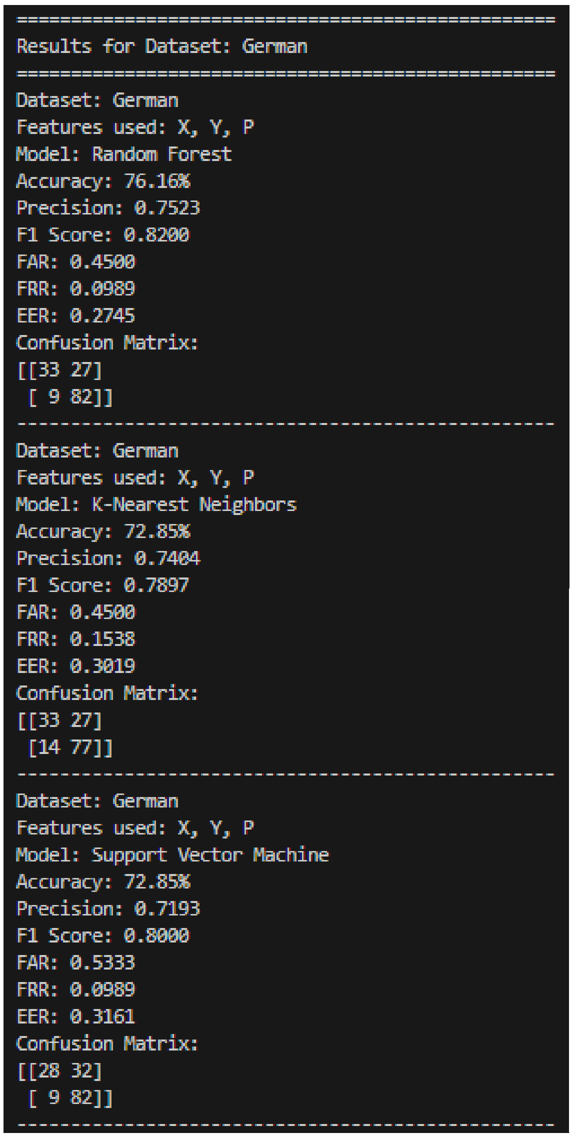


PCA (Train+Test) - Dutch
Principal Component 1
Principal Component 2
0.00
0.25
0.50
0.75
1.00
1.25
1.50
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3

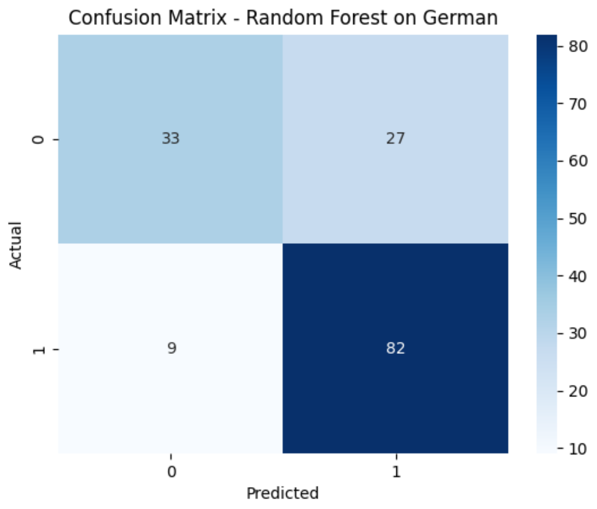
## 5.5 German Dataset

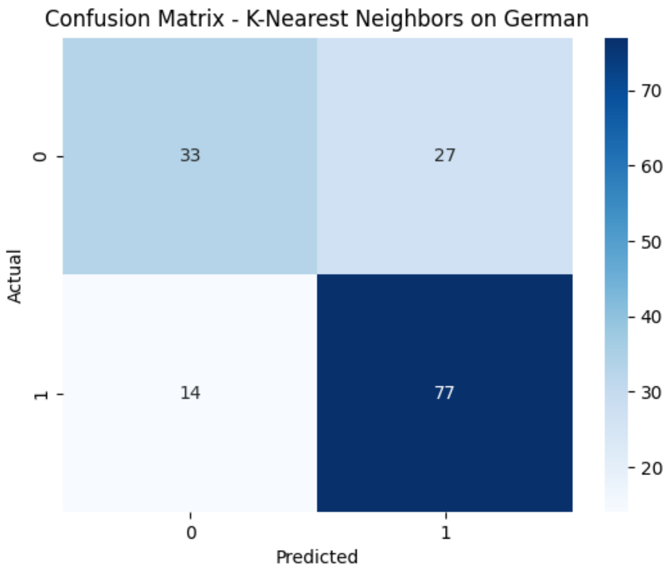
| Model | Accuracy | Precision | F1 Score | FAR | FRR | EER |
|---|---|---|---|---|---|---|
| Random Forest | 76.16% | 0.7523 | 0.8200 | 0.4500 | 0.0989 | 0.2745 |
| K-Nearest Neighbors | 72.85% | 0.7404 | 0.7897 | 0.4500 | 0.1538 | 0.3019 |
| Support Vector Machine | 72.85% | 0.7193 | 0.8000 | 0.5333 | 0.0989 | 0.3161 |

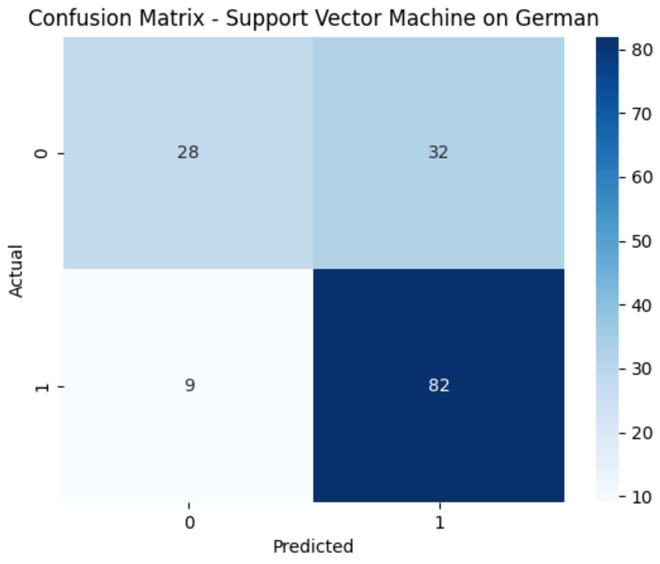
**Highlight**: Overall accuracy and precision were lower for this dataset. Random Forest again came out on top, but high FAR across all models suggests room for feature engineering or data augmentation.

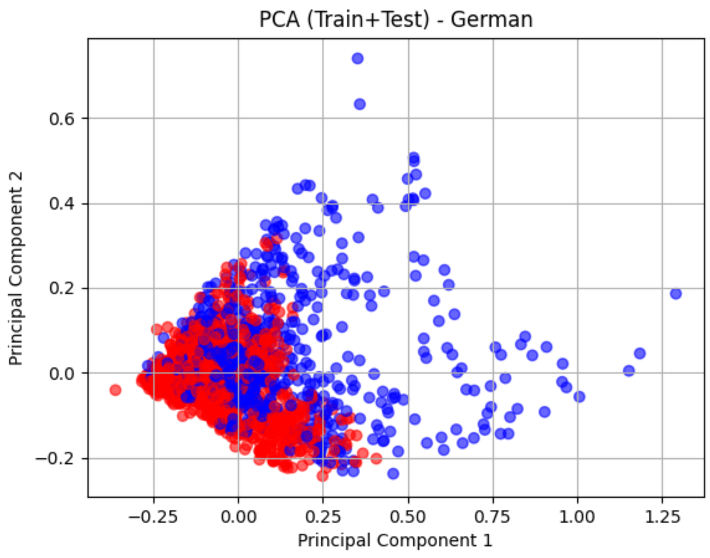


```
==================================================
Results for Dataset: German
==================================================
Dataset: German
Features used: X, Y, P
Model: Random Forest
Accuracy: 76.16%
Precision: 0.7523
F1 Score: 0.8200
FAR: 0.4500
FRR: 0.0989
EER: 0.2745
Confusion Matrix:
[[33 27]
 [ 9 82]]
--------------------------------------------------
Dataset: German
Features used: X, Y, P
Model: K-Nearest Neighbors
Accuracy: 72.85%
Precision: 0.7404
F1 Score: 0.7897
FAR: 0.4500
FRR: 0.1538
EER: 0.3019
Confusion Matrix:
[[33 27]
 [14 77]]
--------------------------------------------------
Dataset: German
Features used: X, Y, P
Model: Support Vector Machine
Accuracy: 72.85%
Precision: 0.7193
F1 Score: 0.8000
FAR: 0.5333
FRR: 0.0989
EER: 0.3161
Confusion Matrix:
[[28 32]
 [ 9 82]]
--------------------------------------------------
```




Confusion Matrix - K-Nearest Neighbors on German
Actual
Predicted
0
1
33
27
14
77
70
60
50
40
30
20


Confusion Matrix - Support Vector Machine on German
Actual
Predicted
0
1
28
32
9
82
80
70
60
50
40
30
20
10


PCA (Train+Test) - German
Principal Component 2
Principal Component 1
0.6
0.4
0.2
0.0
−0.2
−0.25
0.00
0.25
0.50
0.75
1.00
1.25

## 5.6 Trends & Observations

- Top Performer: Random Forest consistently outperformed KNN and SVM across most datasets, showing better accuracy and a balanced FAR/FRR.
- Model Characteristics:
  - SVM often achieved higher precision but had higher FRR, rejecting genuine samples more frequently.
  - KNN offered consistent results but suffered from a higher FAR.
- Dataset Difficulty:
  - MCYT was the most accurate and predictable.
  - German was the most challenging, with the lowest accuracy and highest error rates.
- Feature Influence:
  - Datasets like MCYT and SVC included additional features (al, az) which likely improved model discrimination.

## 5.7 Visual Analysis

Each model-dataset result included:

- Confusion Matrix Heatmaps – to visually assess prediction errors and correct classifications.
- PCA Scatter Plots – to show how well genuine and forged signatures were separated in reduced-dimensional space after scaling.

---

## 6. Conclusion and Future Work

This project successfully explored the use of machine learning algorithms for the task of **online signature verification**, a critical biometric authentication challenge. By evaluating three widely used classifiers—**Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM)**—across five distinct datasets (MCYT, SVC, Chinese, Dutch, German), I developed a robust pipeline that included data preprocessing, feature normalization, model training, performance evaluation, and visual analysis using PCA and confusion matrix heatmaps.

The models were assessed using multiple metrics: **Accuracy, Precision, F1 Score, FAR (False Acceptance Rate), FRR (False Rejection Rate), and EER (Equal Error Rate)**. Among the models tested, **Random Forest consistently outperformed others across most datasets**. It delivered the highest accuracy on the MCYT dataset (93%) and achieved a perfect FRR of 0.0000, meaning no genuine signature was misclassified as forged. This superior performance can be attributed to its ensemble structure, which reduces overfitting and better captures feature interactions.

KNN and SVM showed reliable performance as well, with SVM achieving the highest precision in certain datasets, indicating its effectiveness in minimizing false positives. However, both struggled with balancing FAR and FRR, especially on more challenging datasets like German and SVC.

## Future Work

To enhance this system further:

1. **Time-Series Feature Engineering** – Incorporate features like velocity or acceleration and use DTW to better capture writing dynamics.
2. **Real-Time Verification** – Adapt the system for live signature input to enable deployment in real-world authentication.
3. **Deep Learning Models** – Explore CNNs or RNNs to automatically learn complex temporal patterns in signature data.

This work lays the groundwork for more advanced, scalable, and accurate signature verification systems in digital environments.