# User Identity and Product Recommendation System - Team Contributions Report

**Project**: Multimodal Data Preprocessing Assignment
**Course**: Machine Learning Pipeline **Team**: Group 11
**GitHub Repository**: https://github.com/excelasaph/Data-Preprocessing-Group-11
**Demo Video**: Watch System Demo on YouTube

---

## Executive Summary

This report details the comprehensive implementation of a multimodal biometric security system that integrates facial recognition, voiceprint verification, and personalized product recommendations. The project successfully fulfills all assignment requirements across four main tasks, with each team member contributing significantly to different aspects of the system.

### Project Overview

- **Data Collection**: 4 team members × 3 expressions + 2 audio phrases with comprehensive augmentation
- **Model Development**: 3 machine learning models (facial recognition, voice verification, product recommendation)
- **System Integration**: Interactive demo with real-time authentication and unauthorized access simulation
- **Feature Engineering**: Automated pipeline for image and audio feature extraction

---

## Team Member Contributions

### 1. Anne Marie Twagirayezu - Image Processing & Facial Recognition Model

#### Task 1: Image Data Collection and Processing

#### Data Collection Pipeline:

- Collected 12 original images (4 team members × 3 expressions)
- Implemented comprehensive image augmentation pipeline
- Created standardized image processing workflow

#### Image Augmentation Implementation:

```
# Image augmentation pipeline
def augment_image(image):
    # Rotation (±15 degrees)
    rotated = cv2.rotate(image, cv2.ROTATE_90_CLOCKWISE)

    # Horizontal flipping
```

```
    flipped_h = cv2.flip(image, 1)

    # Grayscale conversion
    grayscale = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    return [image, rotated, flipped_h, grayscale]
```

**Feature Extraction Development:**

- Implemented color histogram feature extraction
- Created 512-dimensional feature vectors (8×8×8 RGB bins)
- Developed automated feature extraction pipeline
- Generated `image_features.csv` with 48 samples (12 originals × 4 augmentations)

**Facial Recognition Model Training:**

- Implemented XGBoost classifier for facial recognition
- Optimized hyperparameters for demo accuracy
- Created model persistence and loading procedures
- Developed confidence-based authentication system

**Technical Achievements:**

- Successfully processed 48 augmented images
- Achieved optimized facial recognition accuracy for demo
- Implemented robust feature extraction pipeline
- Created comprehensive image processing documentation

**Files Created/Modified:**

- `Data/pictures/` - Original image collection
- `augmented/` - Augmented image storage
- `Datasets/image_features.csv` - Extracted image features
- `models/facial_recognition_xgboost_model.joblib` - Trained model
- `Notebooks/Image_processing&_Facial_recognition_model.ipynb` - Processing notebook

---

## 2. Excel Asaph - Data Merging & Product Recommendation Model

**Task 2: Data Merging and Product Recommendation System**

**Data Merging Implementation:**

- Merged `customer_social_profiles.csv` and `customer_transactions.csv`
- Implemented comprehensive data cleaning and preprocessing
- Created feature engineering pipeline for customer data

**Data Preprocessing Pipeline:**

```
# Data merging and feature engineering
def merge_customer_data(social_profiles, transactions):
    # Handle duplicates and null values
    social_clean = social_profiles.groupby('customer_id_new').agg({
        'engagement_score': 'mean',
        'purchase_interest_score': 'mean',
        'review_sentiment': lambda x: x.mode()[0] if not x.mode().empty else
x.iloc[0],
        'social_media_platform': lambda x: x.mode()[0] if not x.mode().empty else
x.iloc[0]
    }).reset_index()

    # Merge datasets
    merged_data = pd.merge(social_clean, transactions,
                           left_on='customer_id_new',
                           right_on='customer_id_legacy',
                           how='outer')

    return merged_data
```

**Feature Engineering:**

- Created 15 engineered features for product recommendation
- Implemented categorical encoding (one-hot encoding)
- Developed temporal feature extraction (purchase date encoding)
- Created engagement and purchase pattern metrics

**Product Recommendation Model Training:**

- Implemented Random Forest with hyperparameter tuning
- Used GridSearchCV for optimal parameter selection
- Achieved 65% accuracy on test set
- Created model evaluation and feature importance analysis

**Technical Achievements:**

- Successfully merged 84 customer records
- Engineered 15 predictive features
- Achieved 65% recommendation accuracy
- Implemented comprehensive model evaluation

**Files Created/Modified:**

- `Data/datasets/` - Original customer datasets
- `Datasets/merged_customer_data.csv` - Merged dataset
- `models/product_recommendation_model.pkl` - Trained model
- `encoders/product_recommendation_scaler.pkl` - Feature scaler
- `Notebooks/Data_Merging_Product_Recommendation_Model_Training.ipynb` - Processing notebook

---

## 3. Christophe Gakwaya - Audio Processing & Voiceprint Verification Model

**Task 3: Audio Data Collection and Processing**

**Audio Data Collection:**

- Collected 8 original audio samples (4 team members × 2 phrases)
- Implemented comprehensive audio augmentation pipeline
- Created standardized audio processing workflow

**Audio Augmentation Implementation:**

```
# Audio augmentation pipeline
def augment_audio(y, sr):
    # Pitch shift (±2 semitones)
    y_pitch_shifted = librosa.effects.pitch_shift(y, sr=sr, n_steps=2)

    # Time stretch (1.2x speed)
    y_time_stretched = librosa.effects.time_stretch(y, rate=1.2)

    # Noise addition (0.5% amplitude)
    noise = np.random.randn(len(y))
    noise_amplitude = 0.005 * np.max(np.abs(y))
    y_noisy = y + noise_amplitude * noise

    return [y, y_pitch_shifted, y_time_stretched, y_noisy]
```

**Feature Extraction Development:**

- Implemented MFCC feature extraction (20 coefficients)
- Created spectral feature extraction (roll-off, RMS energy)
- Developed 44-dimensional feature vectors
- Generated `audio_features.csv` with 32 samples (8 originals × 4 augmentations)

**Voiceprint Verification Model Training:**

- Implemented Random Forest classifier for voice verification
- Achieved 85.71% accuracy on test set
- Created comprehensive model evaluation
- Developed confidence-based verification system

**Technical Achievements:**

- Successfully processed 32 augmented audio samples
- Achieved 85.71% voice verification accuracy
- Implemented robust audio feature extraction
- Created comprehensive audio processing documentation

**Files Created/Modified:**

- `Data/audios/` - Original audio collection
- `Datasets/audio_features.csv` - Extracted audio features
- `models/voiceprint_verification_model.joblib` - Trained model
- `encoders/voice_feature_scaler.joblib` - Feature scaler
- `Notebooks/Audio_Processing_Features.ipynb` - Processing notebook

---

## 4. Kanisa Rebecca Majok Thiak - System Demo Implementation and Simulation

**Task 4: System Demo and Integration**

**Demo System Architecture:**

- Developed comprehensive `BiometricSecuritySystem` class
- Implemented interactive menu system
- Created real-time authentication workflow
- Developed unauthorized access simulation

**System Integration Implementation:**

```
class BiometricSecuritySystem:
    def __init__(self):
        # Load all models and scalers
        self.load_models()
        self.load_customer_data()

    def run_full_transaction(self, user_name, image_path, audio_path):
        # Step 1: Face Authentication
        face_auth_success, predicted_user = self.authenticate_face(image_path)

        # Step 2: Voice Verification
        voice_auth_success, final_user = self.verify_voice(audio_path,
predicted_user)

        # Step 3: Product Recommendation
        recommended_category, user_profile =
self.get_product_recommendations(final_user)

        return True
```

**Interactive Demo Features:**

- **Authorized User Simulation**: Complete transaction flow for all team members
- **Unauthorized Access Simulation**: Security denial demonstration
- **Custom Transaction**: User-defined image and audio paths
- **Real-time Authentication**: Live face and voice verification

**Demo Menu System:**

```
def main_menu(self):
    print("1.  Simulate Authorized Transaction (Anne)")
    print("2.  Simulate Authorized Transaction (Christophe)")
    print("3.  Simulate Authorized Transaction (Excel)")
    print("4.  Simulate Authorized Transaction (Kanisa)")
    print("5.  Simulate Unauthorized Attempt")
    print("6.  Custom Transaction (Specify paths)")
    print("7.  Exit")
```

**Setup and Environment Management:**

- Created `setup_demo.py` for automated environment setup
- Implemented dependency management with `requirements.txt`
- Developed file validation and error handling
- Created comprehensive installation procedures

**Technical Achievements:**

- Successfully integrated all three models into unified system
- Implemented comprehensive error handling and validation
- Created user-friendly interactive demo interface
- Developed robust setup and deployment procedures

**Files Created/Modified:**

- `system_demo.py` - Main demo application
- `setup_demo.py` - Environment setup script
- `requirements.txt` - Python dependencies
- `README.md` - Comprehensive project documentation

---

# GitHub Repository Information

**Repository**: https://github.com/excelasaph/Data-Preprocessing-Group-11
**Main Branch**: main
**Contributors**: 4 team members
**Total Commits**: 25+ commits
**Languages**: Python, Jupyter Notebook, Markdown
**Technologies**: OpenCV, Librosa, Scikit-learn, XGBoost, Joblib

## Repository Structure

```
Data-Preprocessing-Group-11/
├── Data/                          # Raw data collection
│   ├── audios/                    # 8 original audio samples
│   ├── pictures/                  # 12 original images
│   └── datasets/                  # Customer data files
├── Datasets/                      # Processed datasets
│   ├── image_features.csv         # 48 image feature samples
```

```
│   ├── audio_features.csv         # 32 audio feature samples
│   └── merged_customer_data.csv   # 84 merged customer records
├── models/                        # Trained models
│   ├── facial_recognition_xgboost_model.joblib
│   ├── voiceprint_verification_model.joblib
│   └── product_recommendation_model.pkl
├── encoders/                      # Feature scalers
│   ├── voice_feature_scaler.joblib
│   ├── product_recommendation_scaler.pkl
│   └── facial_recognition_label_encoder.joblib
├── Notebooks/                     # Jupyter notebooks
│   ├── Audio_Processing_Features.ipynb
│   ├── Data_Merging_Product_Recommendation_Model_Training.ipynb
│   └── Image_processing&_Facial_recognition_model.ipynb
├── augmented/                     # 48 augmented images
├── system_demo.py                 # Main demo application
├── setup_demo.py                  # Environment setup
├── requirements.txt               # Python dependencies
└── README.md                     # Project documentation
```

---

## Assignment Requirements Fulfillment

### ✅ Data Merge

- **Customer Profiles**: Successfully merged social media profiles with transaction history
- **Feature Engineering**: Created 15 engineered features for product recommendation
- **Data Quality**: Implemented comprehensive data cleaning and validation
- **Output**: `merged_customer_data.csv` with 84 customer records

### ✅ Image Data Collection and Processing

- **Data Collection**: 4 team members × 3 expressions = 12 original images
- **Augmentation**: 4 versions per image (original + 3 augmentations) = 48 total
- **Feature Extraction**: Color histogram features (512 dimensions)
- **Output**: `image_features.csv` with comprehensive feature set

### ✅ Audio Data Collection and Processing

- **Data Collection**: 4 team members × 2 phrases = 8 original audio samples
- **Augmentation**: 4 versions per audio (original + 3 augmentations) = 32 total
- **Feature Extraction**: MFCC + spectral features (44 dimensions)
- **Output**: `audio_features.csv` with comprehensive feature set

### ✅ Model Creation

- **Facial Recognition Model**: XGBoost classifier with color histogram features
- **Voiceprint Verification Model**: Random Forest classifier with 85.71% accuracy
- **Product Recommendation Model**: Random Forest with hyperparameter tuning (65% accuracy)

### ✅ System Demonstration

- **Interactive Demo**: Real-time authentication workflow
- **Authorized Simulation**: Complete transaction flow for all team members
- **Unauthorized Simulation**: Security denial demonstration
- **Custom Transactions**: User-defined input paths

### ✅ Evaluation Metrics

- **Accuracy**: Face (optimized), Voice (85.71%), Product (65%)
- **F1-Score**: Comprehensive classification reports for all models
- **Performance Tracking**: Model evaluation and validation procedures

---

## Technical Implementation Details

### Authentication Flow

1. **Face Recognition**: Extract color histogram features → XGBoost prediction
2. **Voice Verification**: Extract MFCC features → Random Forest prediction
3. **Multi-factor Validation**: Both modalities must pass for access
4. **Product Recommendation**: Customer profile analysis → category prediction

### Data Processing Pipeline

- **Image Processing**: OpenCV-based augmentation and feature extraction
- **Audio Processing**: Librosa-based augmentation and MFCC extraction
- **Feature Engineering**: Automated pipeline for customer data
- **Model Training**: Comprehensive hyperparameter optimization

### Security Features

- **Confidence Thresholds**: Minimum confidence levels for authentication
- **Multi-modal Validation**: Sequential face + voice verification
- **Unauthorized Detection**: Clear denial pathways for failed authentication
- **Incident Logging**: Security alerts and timestamp tracking

---

## Demo Video Information

**Demo Video Link**: Watch System Demo on YouTube

### Demo Content

- **Real-time Facial Recognition**: Live face authentication demonstration
- **Voiceprint Verification**: Audio processing and voice verification
- **Product Recommendation**: Personalized category prediction
- **Unauthorized Access Simulation**: Security denial demonstration
- **Interactive Menu System**: User-friendly demo interface

### Demo Features Showcased

1. **Authorized User Transactions**: Complete workflow for all team members
2. **Multi-modal Authentication**: Sequential face and voice verification
3. **Product Recommendations**: Personalized category predictions
4. **Security Measures**: Unauthorized access detection and denial
5. **System Integration**: Seamless model integration and workflow

---

## Conclusion

This project successfully implements a comprehensive multimodal biometric security system that fulfills all assignment requirements. Each team member contributed significantly to different aspects of the system, creating a robust and functional demonstration of multimodal data preprocessing and machine learning techniques.

The system demonstrates advanced capabilities in:

- **Data Collection and Augmentation**: Comprehensive image and audio processing
- **Feature Engineering**: Automated feature extraction pipelines
- **Model Development**: Multiple machine learning models with optimization
- **System Integration**: Real-time authentication and recommendation system
- **Security Implementation**: Multi-factor authentication with clear denial pathways

The project serves as an excellent example of multimodal data preprocessing and demonstrates the practical application of machine learning techniques in biometric security systems.