**FINGERPRINT DATASET ANALYSIS AND PROCESSING DOCUMENTATION**

**BY GROUP P**

| **NAME** | **REGISTRATION NO.** | **STUDENT NUMBER** |
| --- | --- | --- |
| **BARISIGARA SIMON** | **21/U/12624/PS** | **2100712624** |
| **NAMAYANJA PATRICA LINDA** | **21/U/11024/PS** | **2100711024** |
| **SHEMA COLLINS** | **21/X/20205/PS** | **2100720205** |
| **AINAMAANI DOUGLAS BAGAMBE** | **21/U/3791** | **2100703792** |
| **TUSIIME FORTUNATE AGNES** | **21/U/07293/EVE** | **2100707293** |

**Abstract:** Fingerprint recognition is a mature biometric technique for identification or authentication application. First, an overview of the state of the art in fingerprint recognition is presented, including current issues and challenges. Fingerprint dataset and evaluation campaigns are also summarized. Two research systems are compared within the proposed framework. The evaluated systems follow different approaches for fingerprint processing and are discussed in detail.

1. Introduction:Fingerprint recognition is a well-established biometric identification technique that has found extensive applications in security systems, law enforcement, and access control. The uniqueness and permanence of fingerprints make them an ideal candidate for accurate and reliable identification.

In this model, the fingerprint dataset was used and comprises real images and Altered images that are distorted in three different levels (Easy, Medium, Hard). The images also contain labels for each image classifying it as labeled data hence supervised learning techniques have been applied in the model.

This concept paper outlines a comprehensive methodology for fingerprint matching using advanced techniques, aiming to enhance accuracy and efficiency in real-world scenarios.

1. Problem Statement:Fingerprint matching involves comparing a query fingerprint against a database of enrolled fingerprints to determine whether a match exists. The challenge lies in developing methodologies that can handle variations in fingerprint patterns due to factors such as image quality, orientation, and deformation.
2. Key Objectives:The primary objective of this research is to develop a fingerprint matching system that exhibits significantly improved accuracy compared to traditional methods. This is achieved by leveraging advanced feature extraction techniques that capture intricate details in fingerprint patterns. By utilizing minutiae-based features, ridge frequency analysis, and texture-based attributes, the system can discriminate between fingerprints with higher precision, even when dealing with variations caused by image quality and orientation.

Efficient and Fast Matching Process: Another crucial objective is to design a fingerprint matching process that combines both speed and accuracy.

# Methodology:

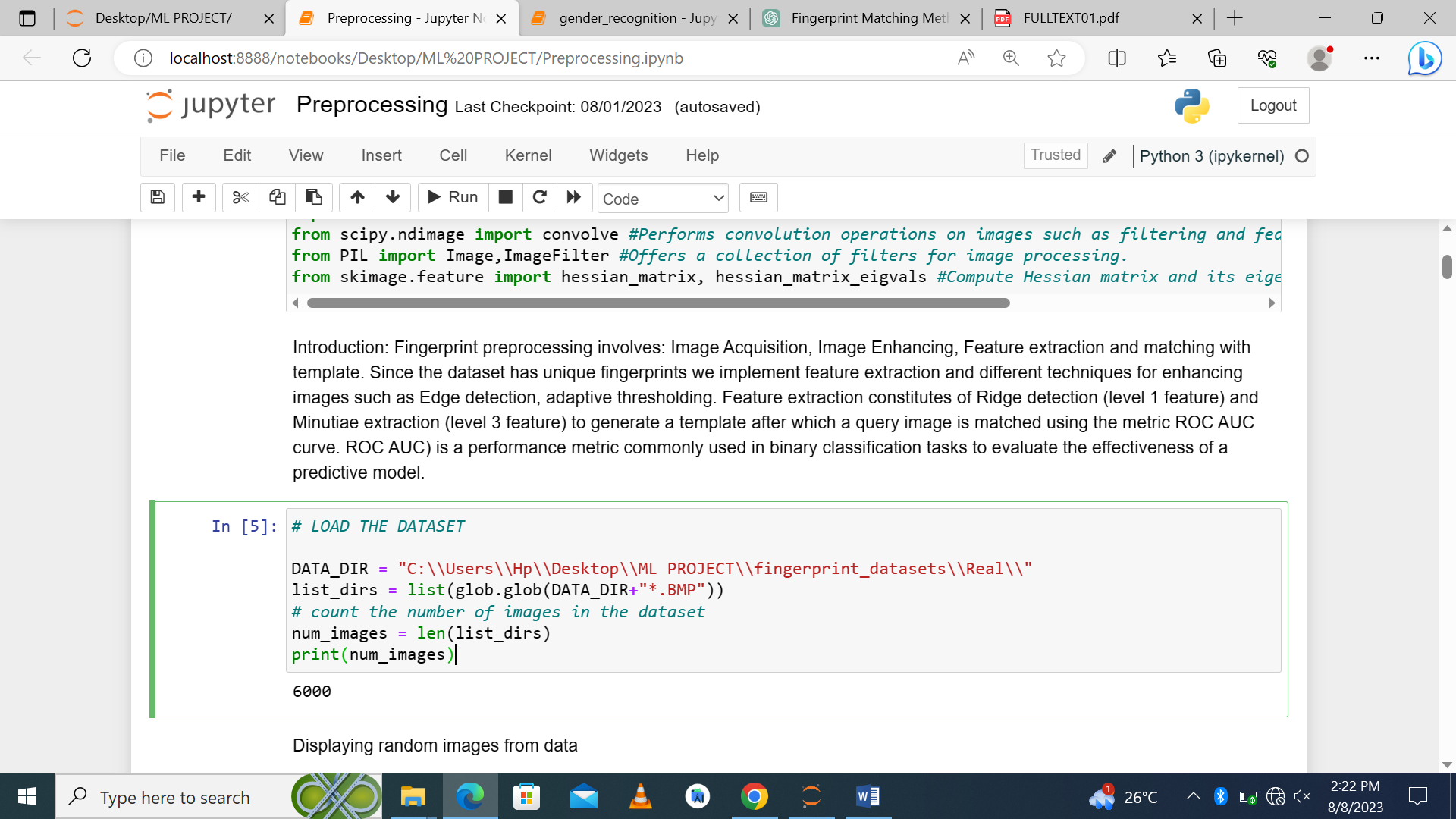
## Data Collection and Preprocessing:

Acquire a diverse and representative dataset of fingerprint images to ensure robustness of the matching system.

Apply preprocessing techniques like noise reduction, image enhancement, and normalization to improve the quality and consistency of fingerprint images.

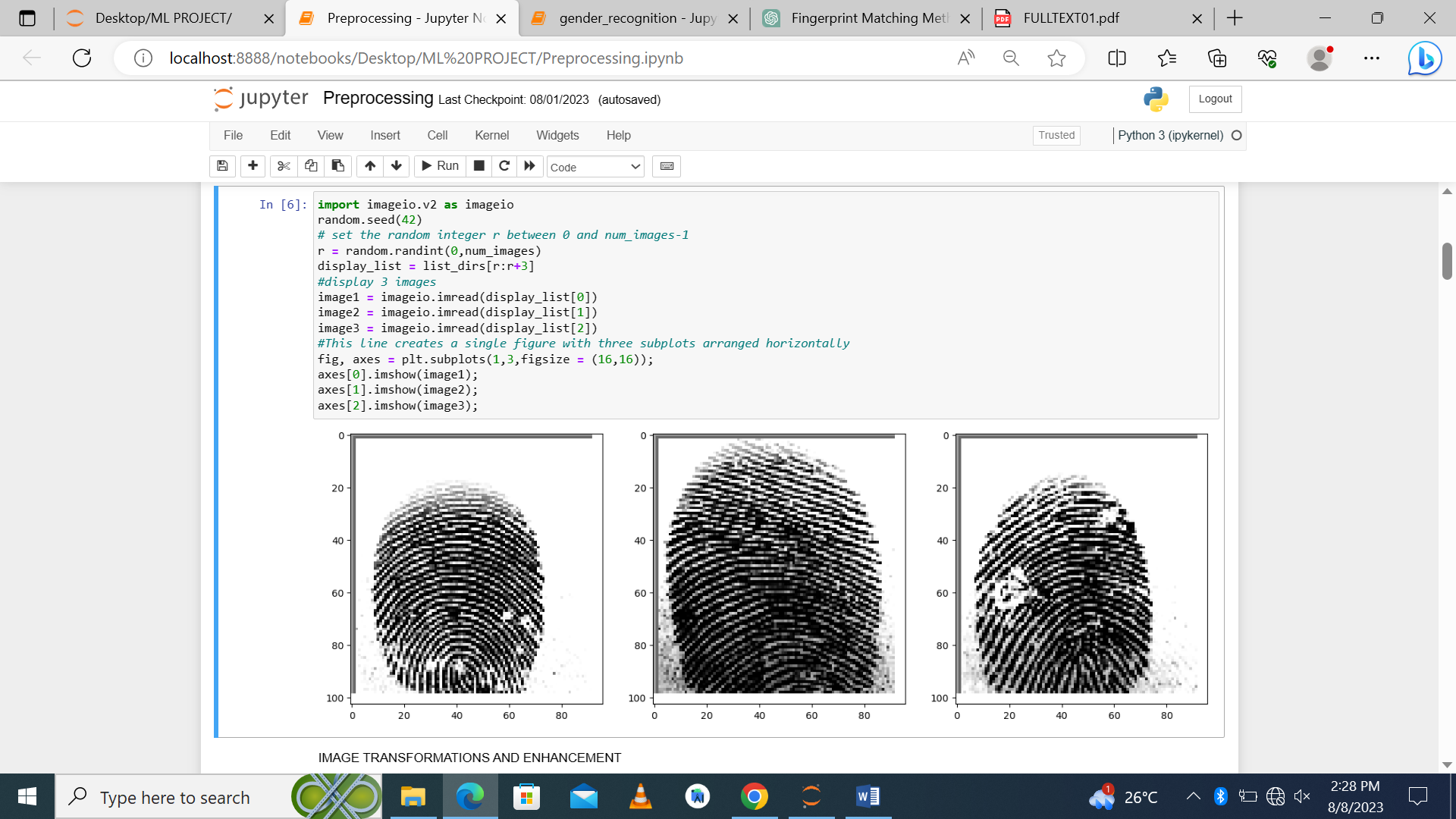
1. Feature Extraction: Fingerprint preprocessing involves: Image Acquisition, Image Enhancing, Feature extraction and matching with template. Since the dataset has unique fingerprints we implement feature extraction and different techniques for enhancing images such as Edge detection, adaptive threshold. Feature extraction consists of Ridge detection (level 1 feature) and Minutiae extraction (level 3 feature) to generate a template after which a query image is matched using the metric ROC AUC curve. ROC AUC) is a performance metric commonly used in binary classification tasks to evaluate the effectiveness of a predictive model.

Image acquisition involves extracting the dataset and loading it into the virtual environment to be used.



*Figure* ***Error! No text of specified style in document.****.1 shows code to load images in Real directory from dataset*

1. Displaying Random Images

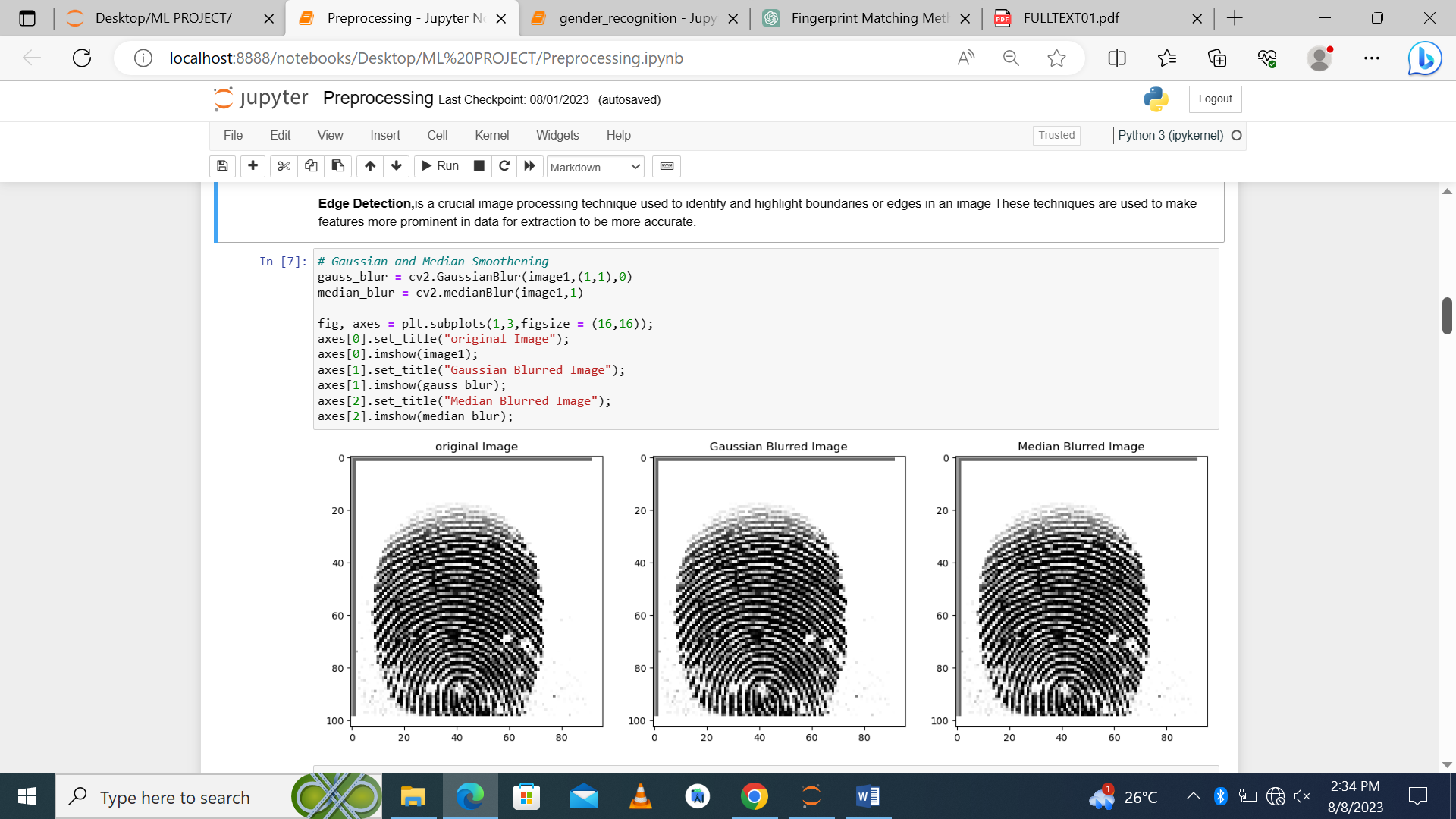


*Figure* ***1.Error! No text of specified style in document.****.2 shows 3 random images from real directory*

## **Image transformations and enhancement**

**Image Smoothening** to reduce noise and remove small details in an image while preserving its overall structure. Techniques used in smoothening; The choice of a smoothing technique depends on the specific application and the characteristics of the noise present in the image.

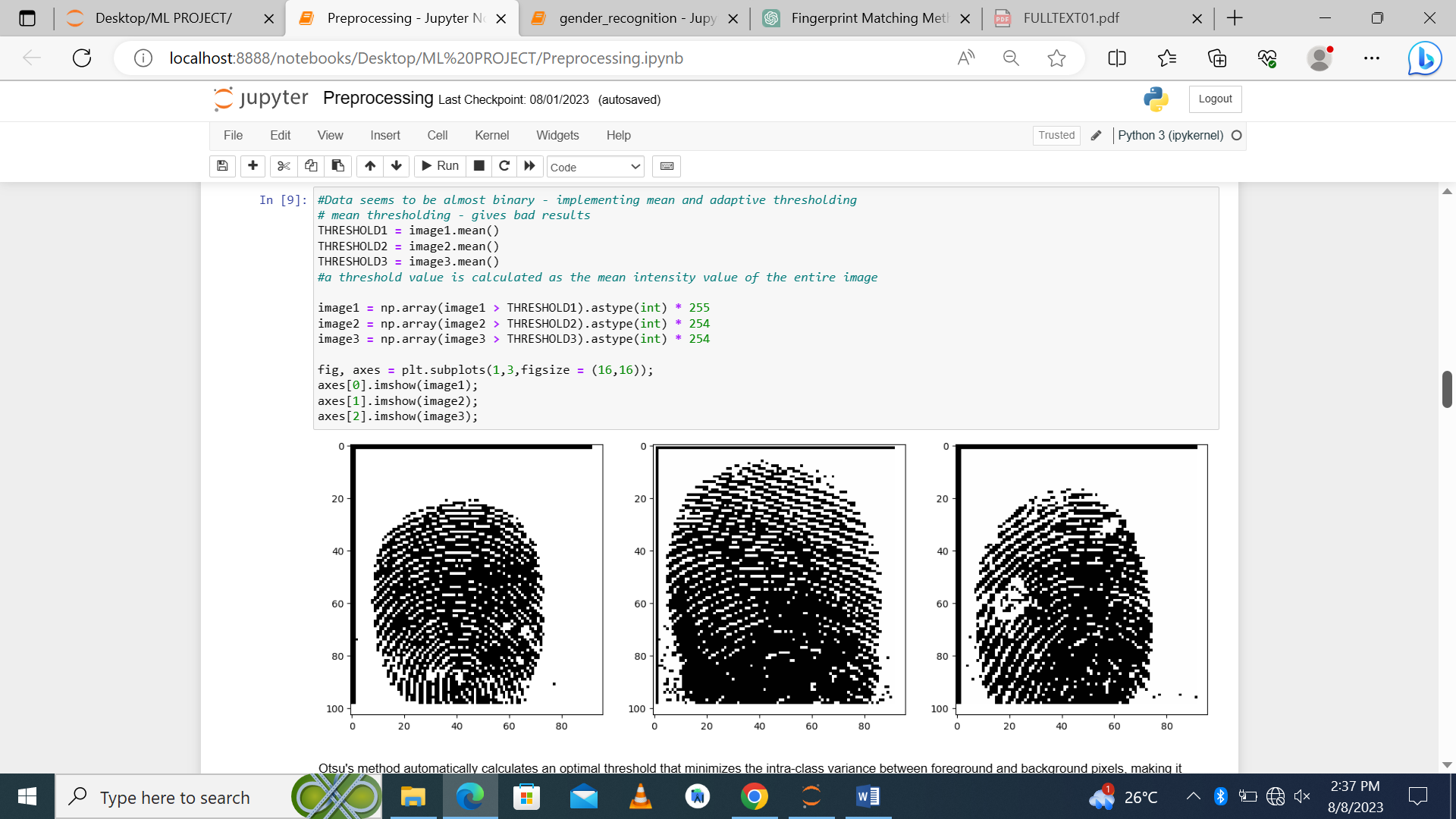
**Gaussian Smoothing**: It applies a Gaussian kernel to the image, which is a bell-shaped function. The convolution operation with the Gaussian kernel effectively blurs the image, reducing the effect of noise**. Median Filtering:** It replaces each pixel in the image with the median value of the pixels in its neighborhood. Median filtering is effective in removing salt-and-pepper noise while preserving edges**. Average Smoothing (Mean Filtering):** It replaces each pixel in the image with the average value of the pixels in its neighborhood. This method is simple and can be effective for reducing random noise.



**Threshold,** is a fundamental image processing technique used to convert a grayscale image into a binary image. In a binary image, each pixel is either classified as "foreground" (typically represented by a white pixel, with a value of 1) or "background" (usually represented by a black pixel, with a value of 0), based on a specific threshold value. It is useful in image segmentation.

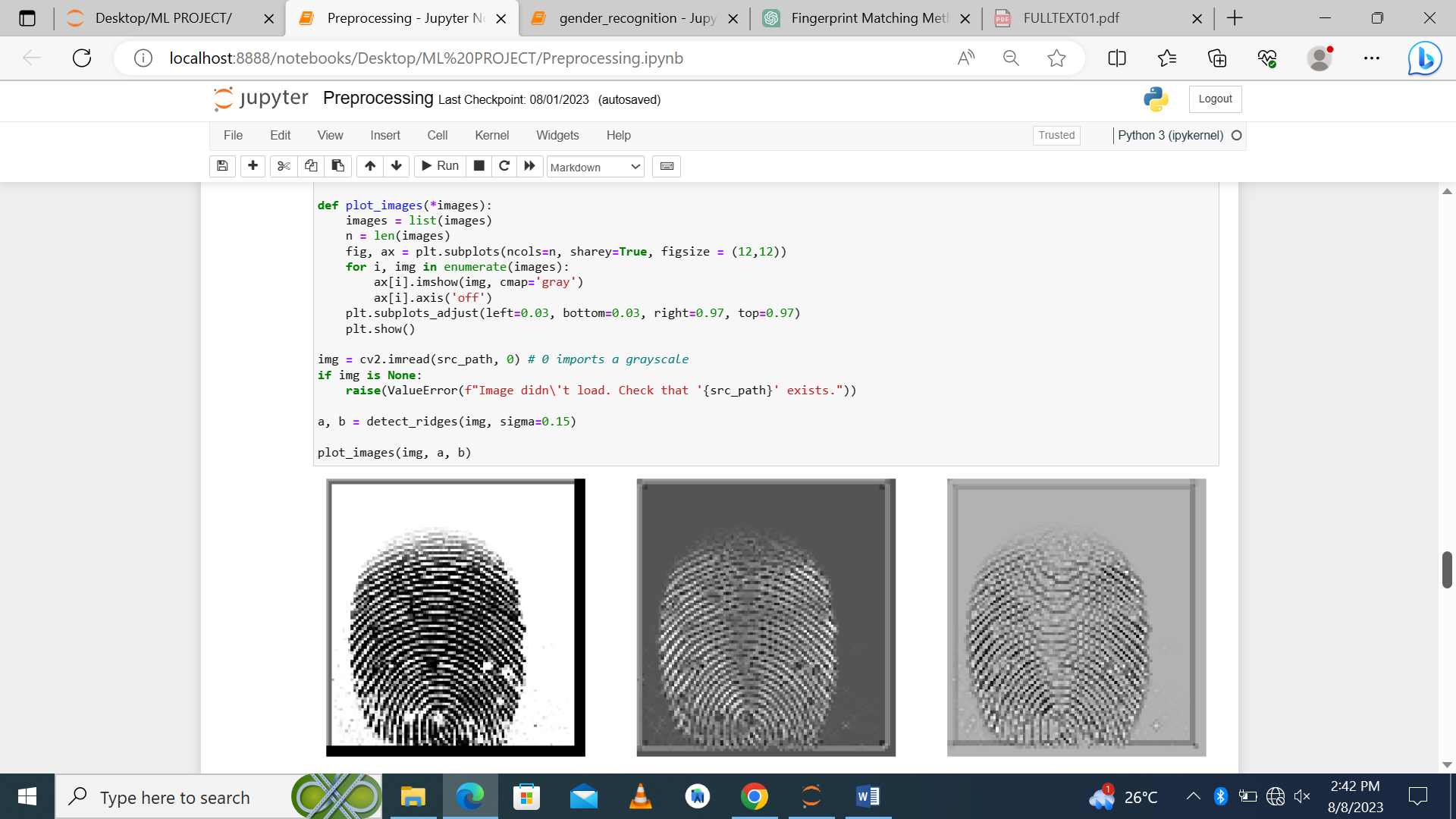
Global Threshold: A single threshold value is applied to the entire image, assuming the foreground and background have distinct intensity values.

Adaptive threshold: Different threshold values are applied to different regions of the image, taking local intensity variations into account. Adaptive threshold is suitable for images with varying lighting conditions. Binary Threshold: The simplest form of threshold, where a fixed threshold value is manually set to segment the image.

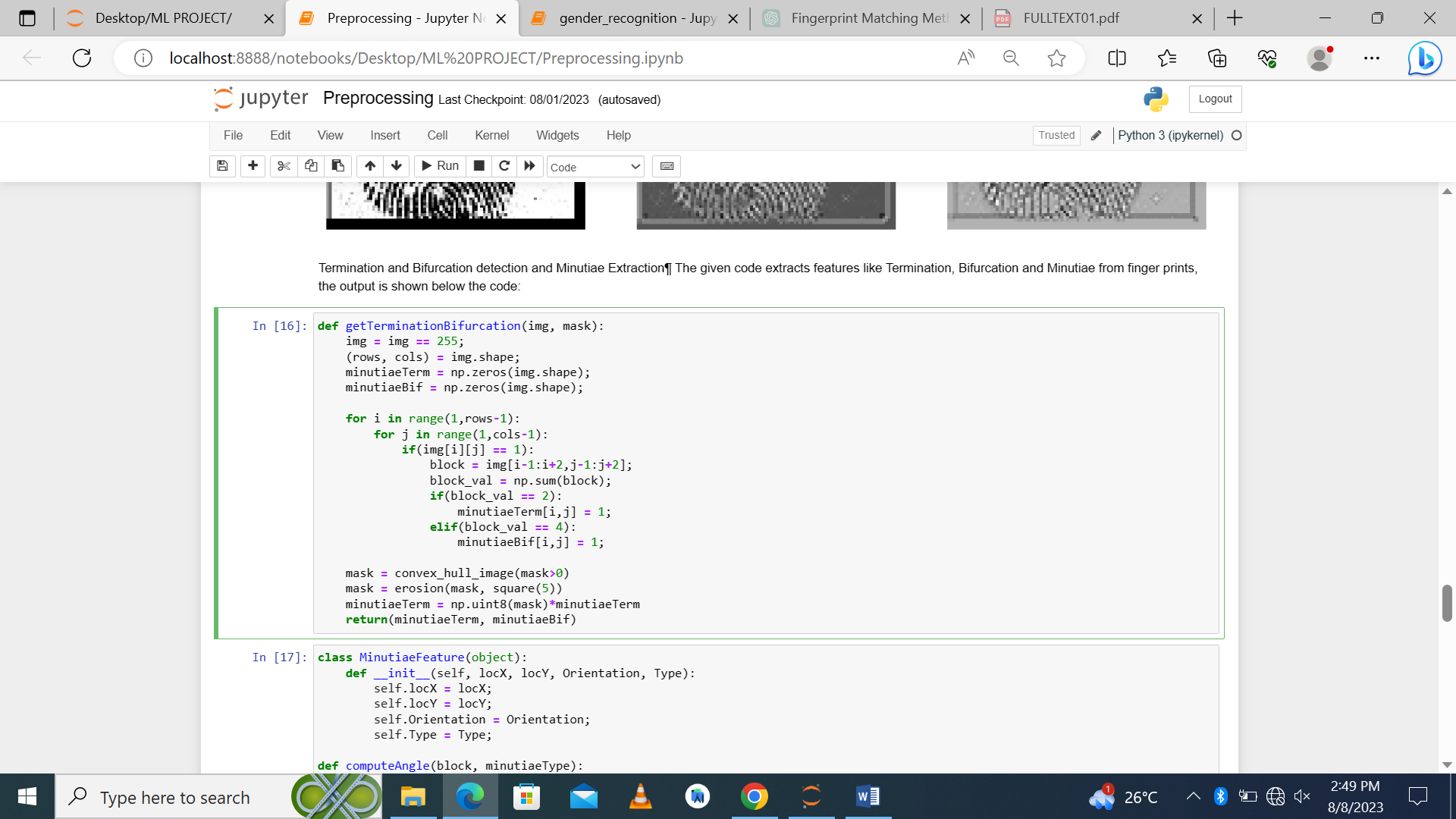


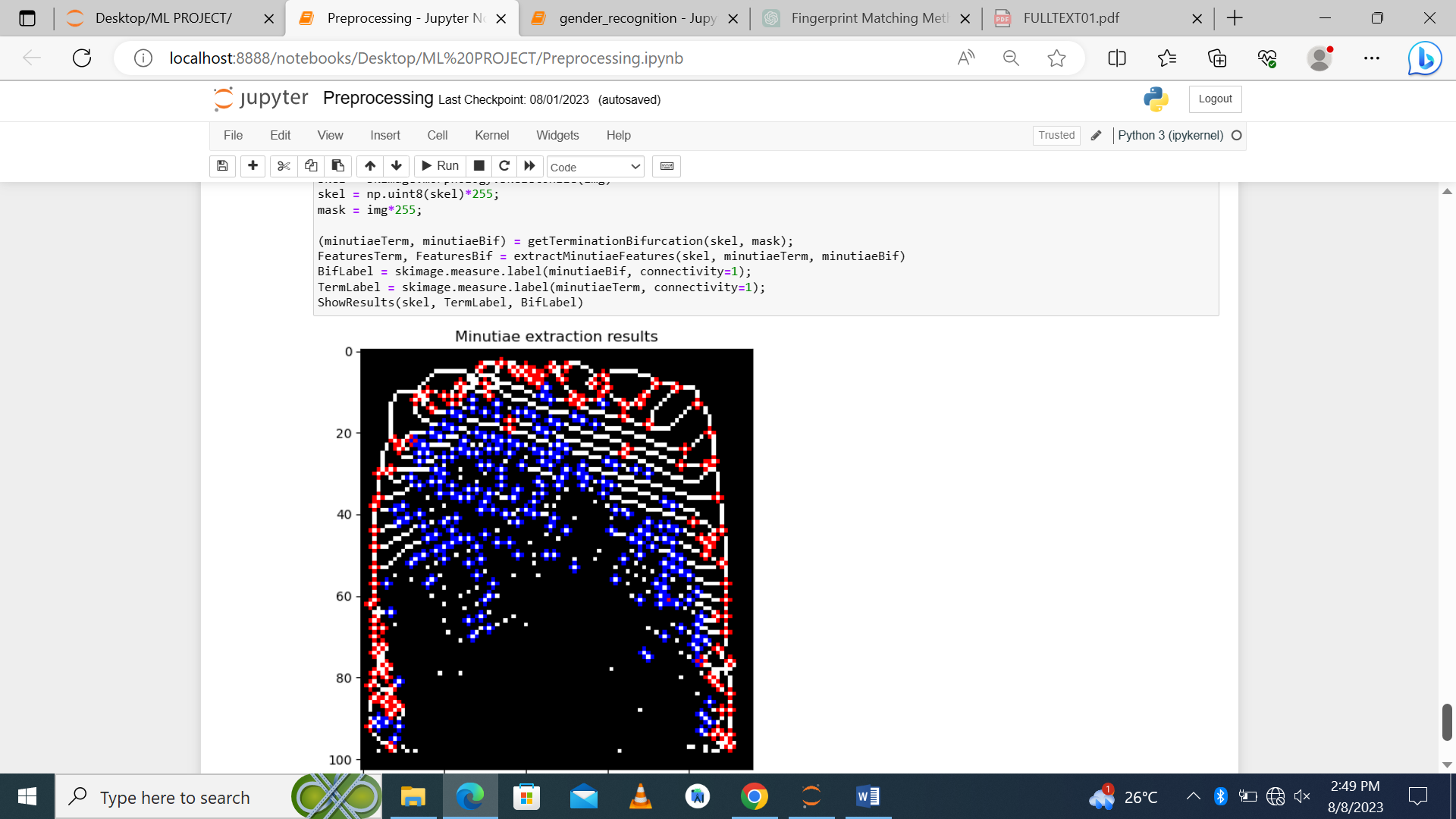
*Figure 1.3 code snippet shows binarization using threshold*

**Detecting ridges**: In a grayscale image using the Hessian matrix. The Hessian matrix represents the local structure of the image intensity. Sigma controls the scale of the Hessian calculation, in this case, maxima\_ridges represents the maximum curvature, which indicates ridge-like structures, and minima\_ridges represents the minimum curvature, which indicates valley-like structures.



**Ridge detection:** Termination and Bifurcation detection and Minutiae Extraction¶ The given code extracts features like Termination, Bifurcation and Minutiae from fingerprints, the output is shown below the code:

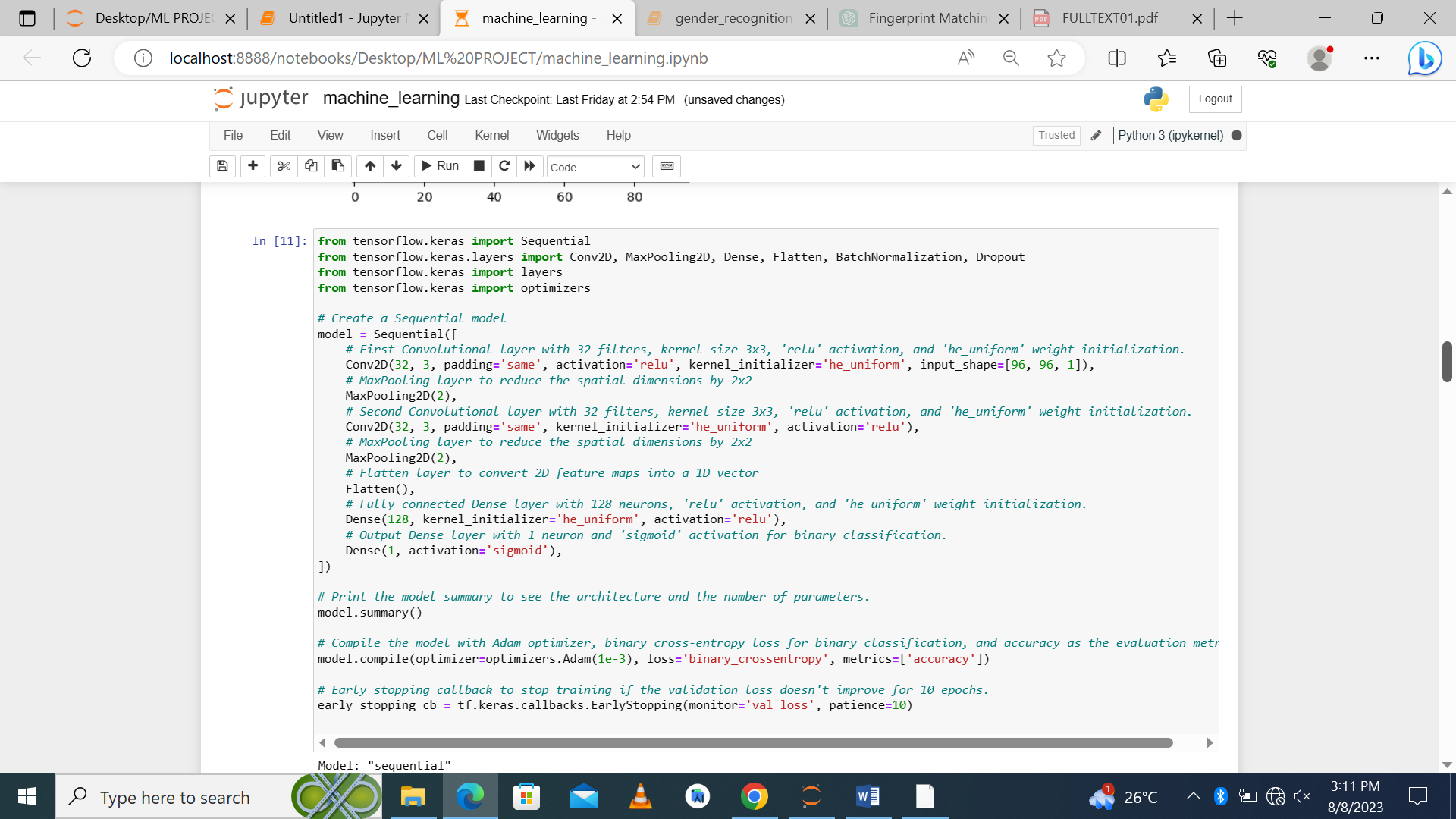
****

****

Additionally, employ advanced techniques like ridge frequency analysis and texture-based features to capture finer details and patterns within the fingerprint.

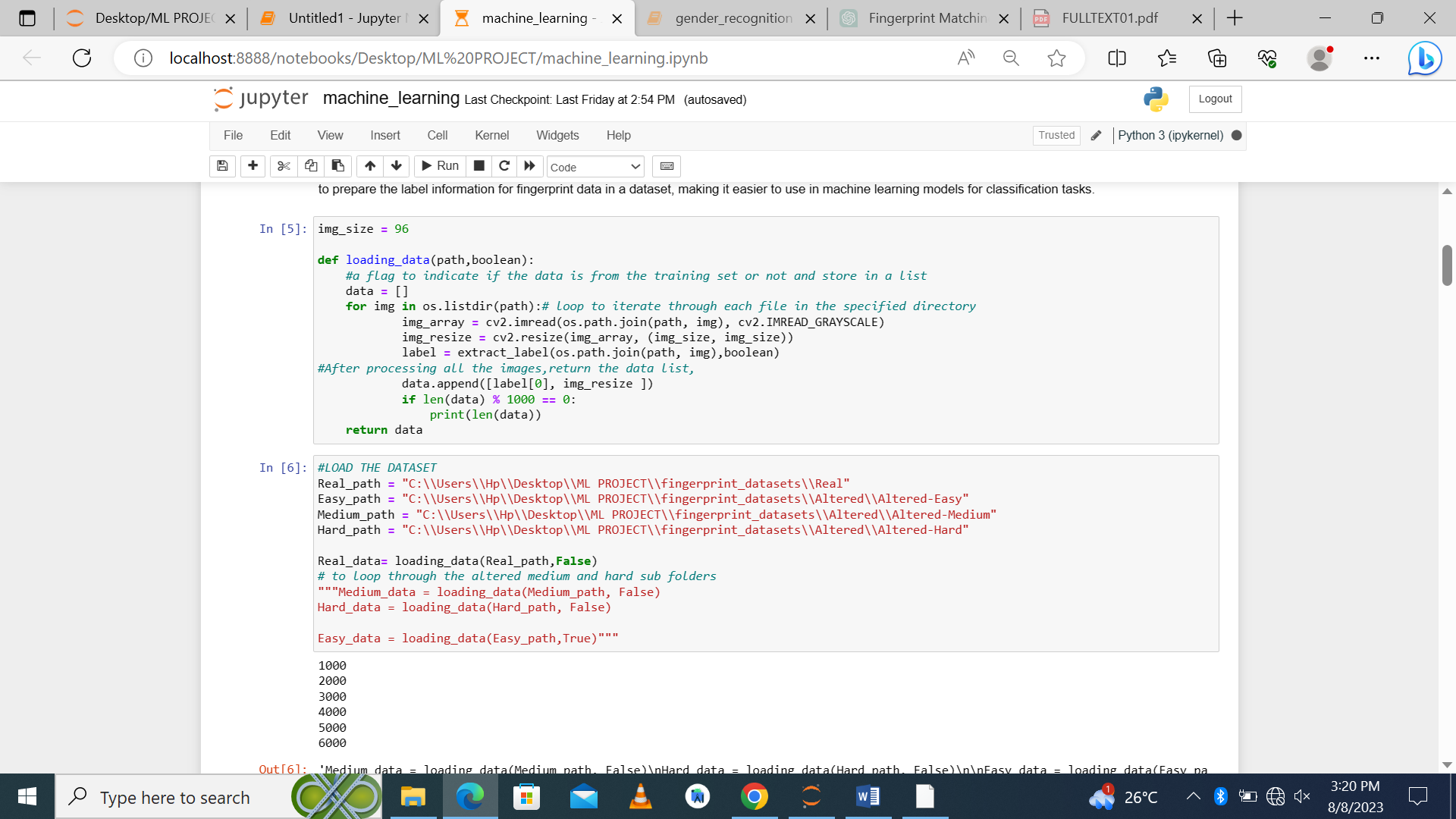
# Feature Representation:

Convert extracted features into a suitable representation that can be used for matching. Explore techniques such as local feature descriptors (e.g., SIFT, SURF) or deep learning-based embedding to create compact yet discriminative representations.

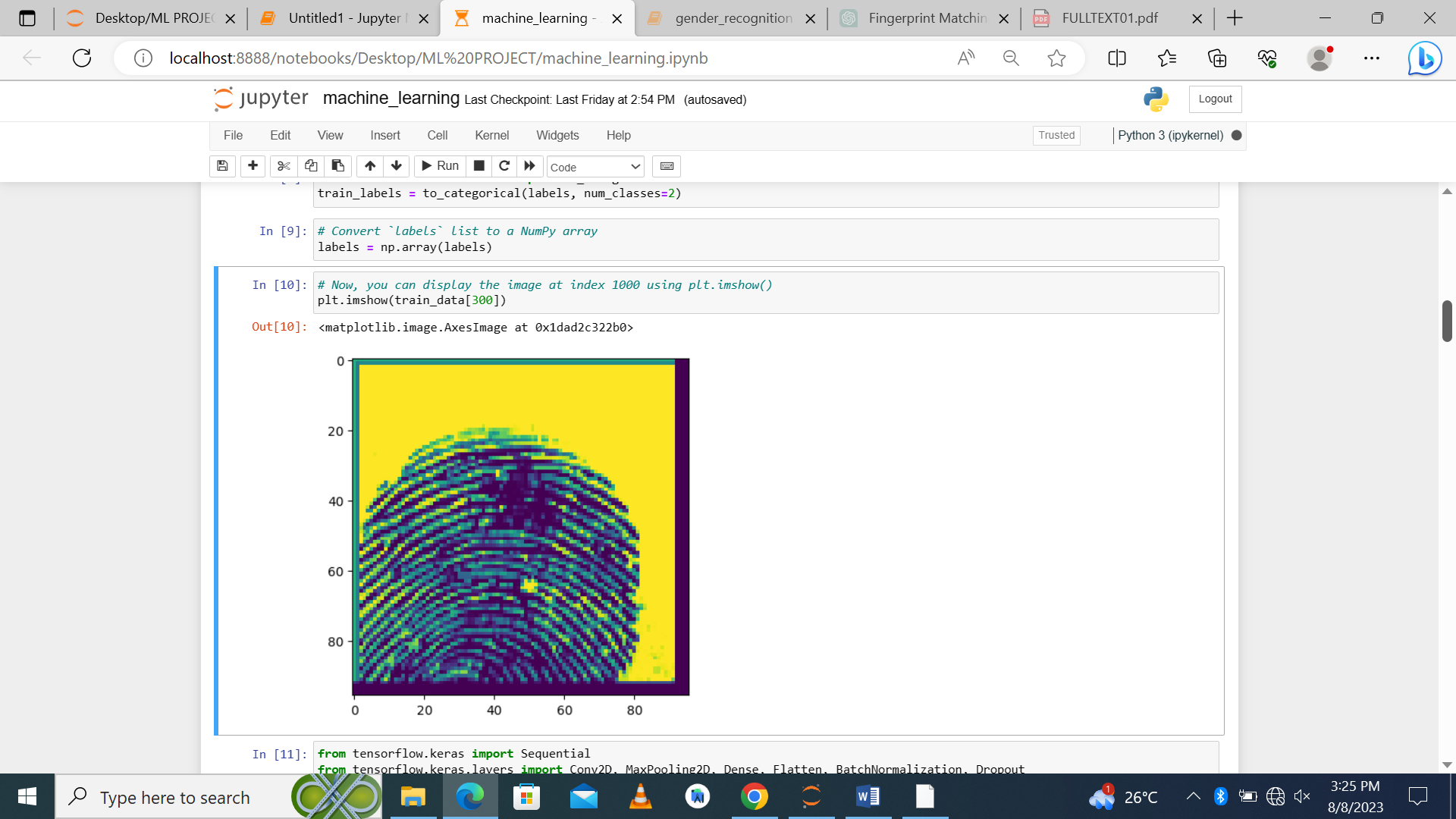


1. **Split the dataset:** the dataset is split into 30% for training data and 70% for testing. 25% of the training data is used to evaluate the model.

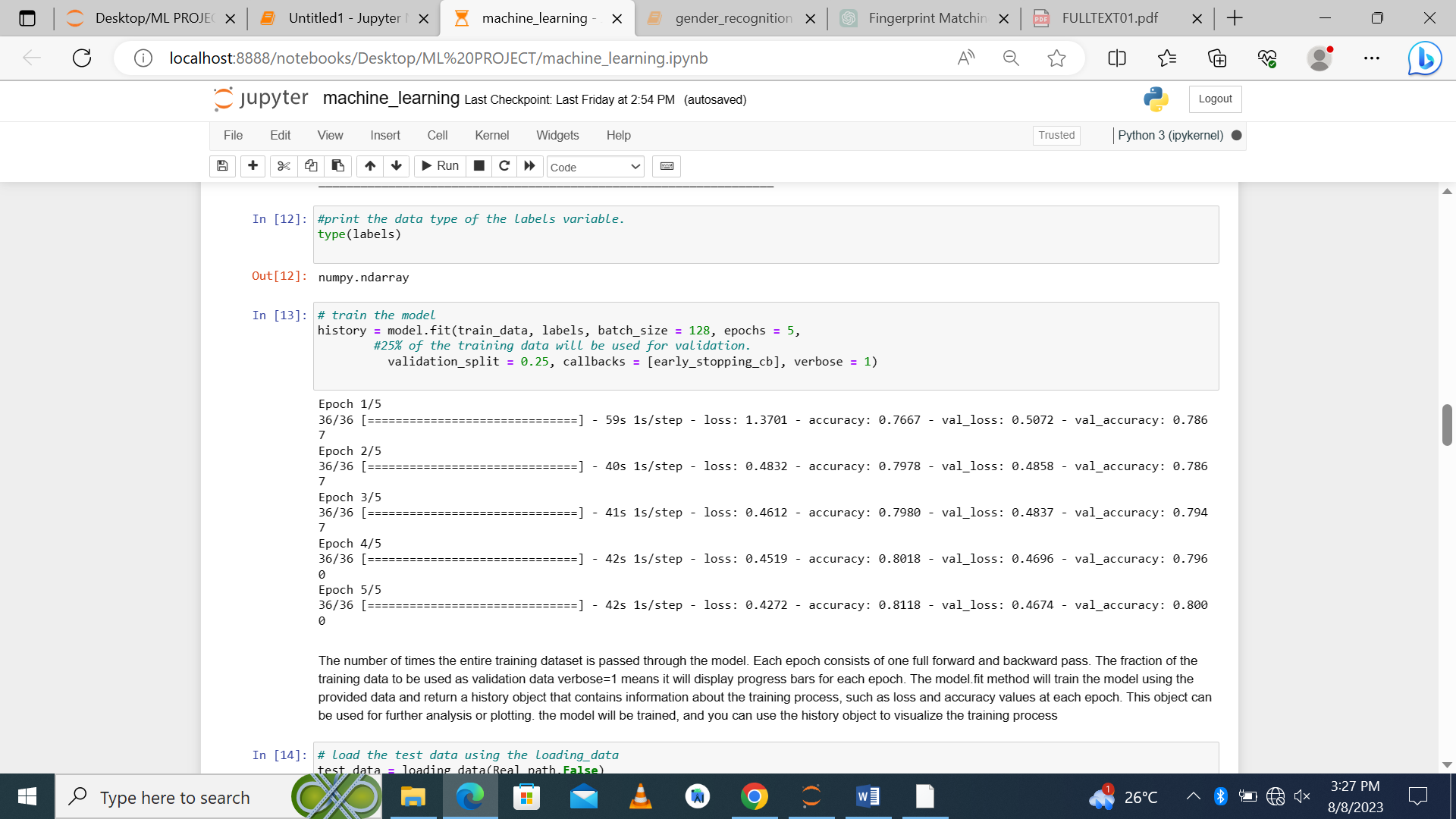
**Training:**  Before the model is trained, the images in the training list are converted into binary format that the model can easily understand, convert them into a numpy array, fit into a uniform size that can be integrated to fit into the model.



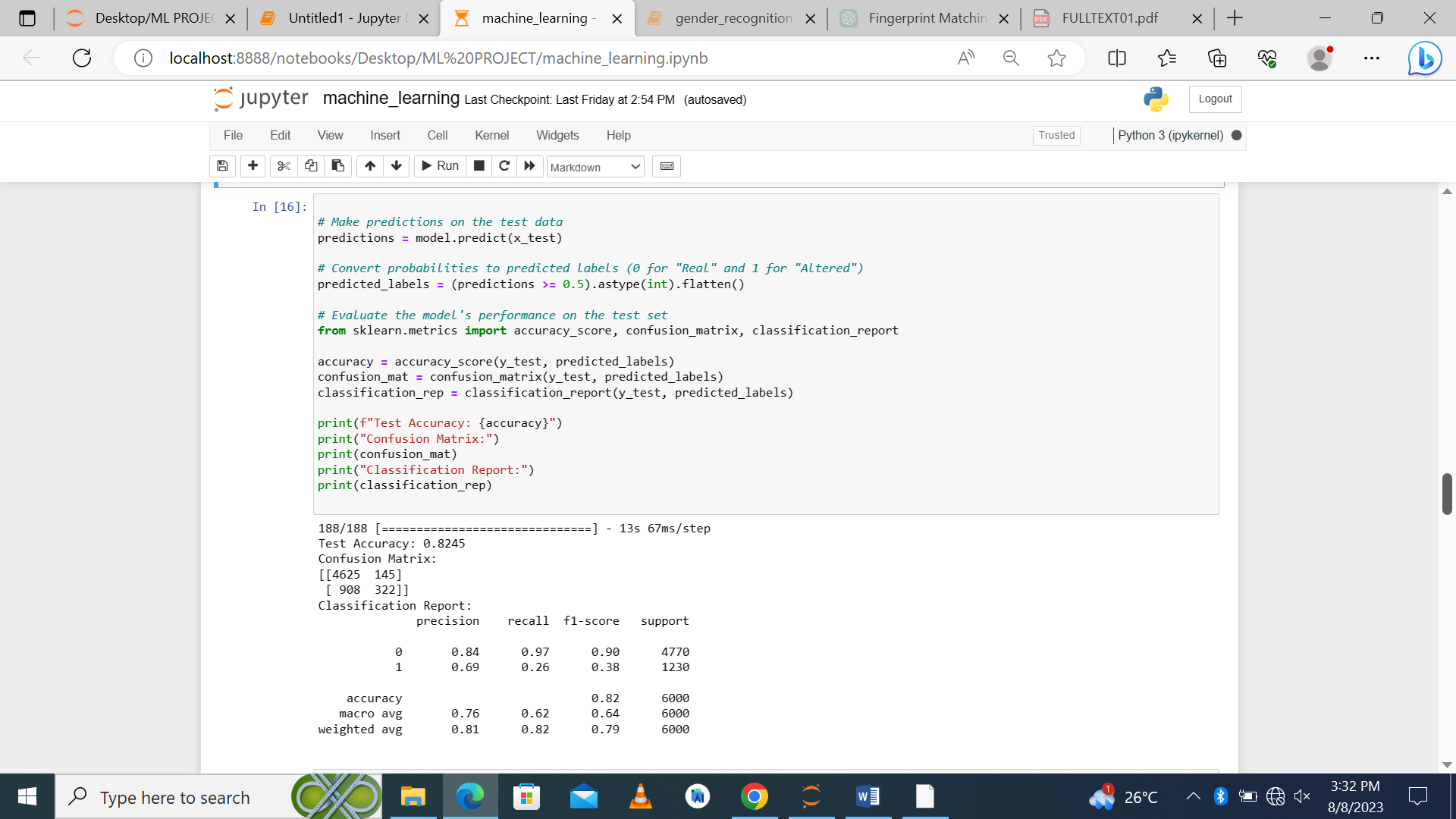




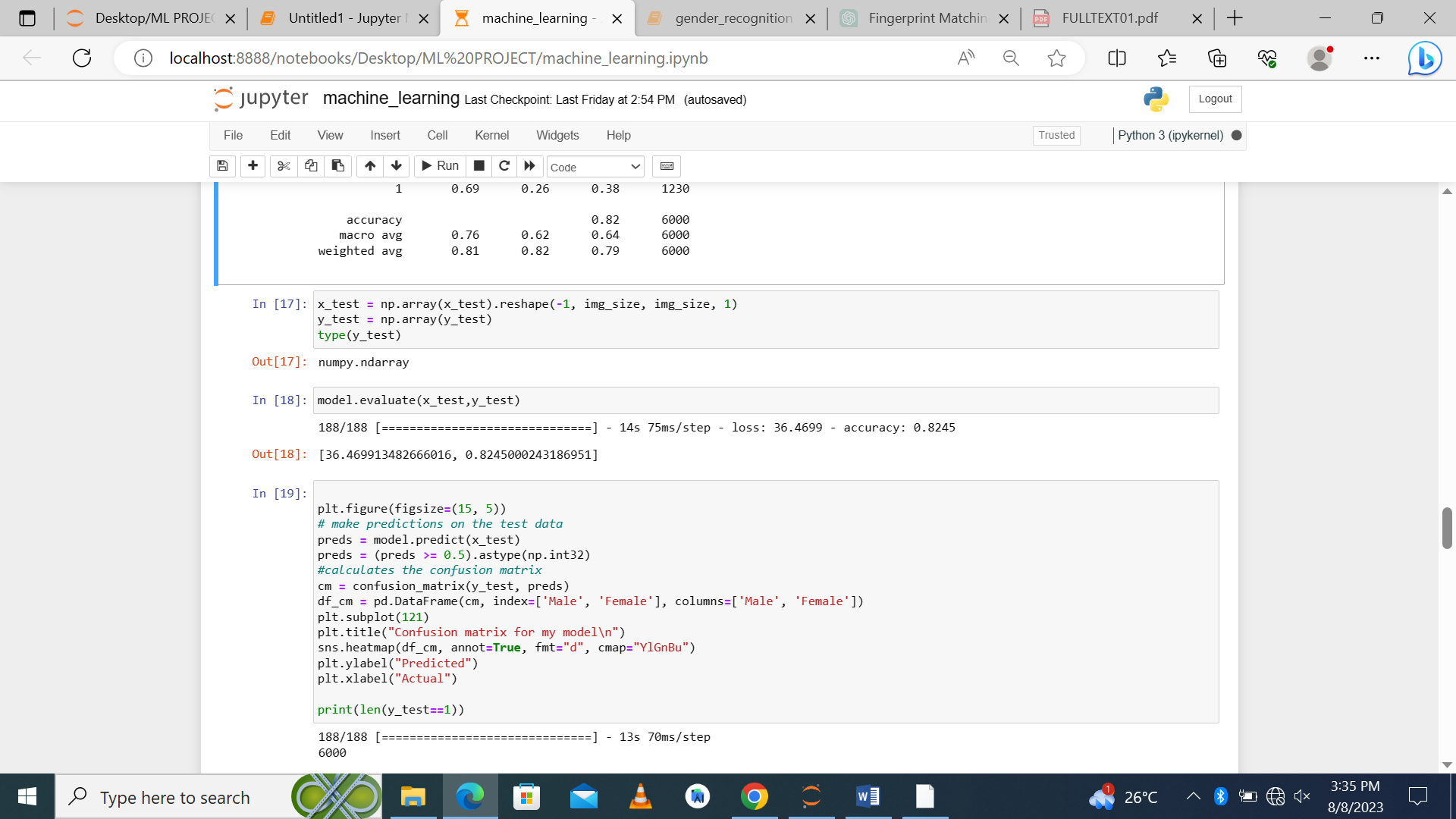
1. **Train the model:** The number of times the entire training dataset is passed through the model. Each epoch consists of one full forward and backward pass. The fraction of the training data to be used as validation data verbose=1 means it will display progress bars for each epoch. The model. fit method will train the model using the provided data and return a history object that contains information about the training process, such as loss and accuracy values at each epoch. This object can be used for further analysis or plotting. the model will be trained, and you can use the history object to visualize the training process



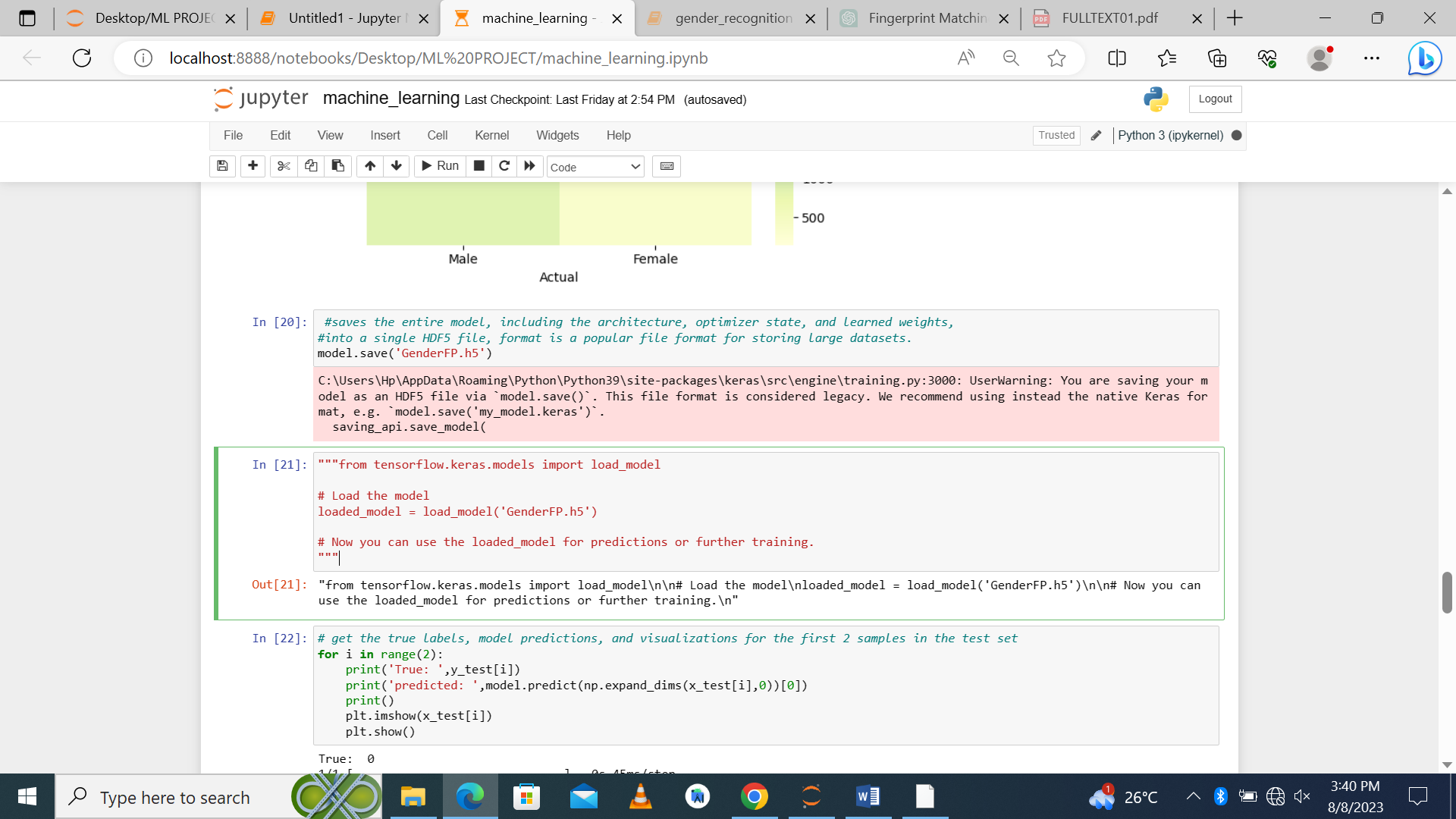
**use the trained model**: to predict the labels of the test data and evaluate the model's performance on the test set.



1. **Test and Evaluate the model:** the model can be trained again to obtain better accuracy, precision and f1-score

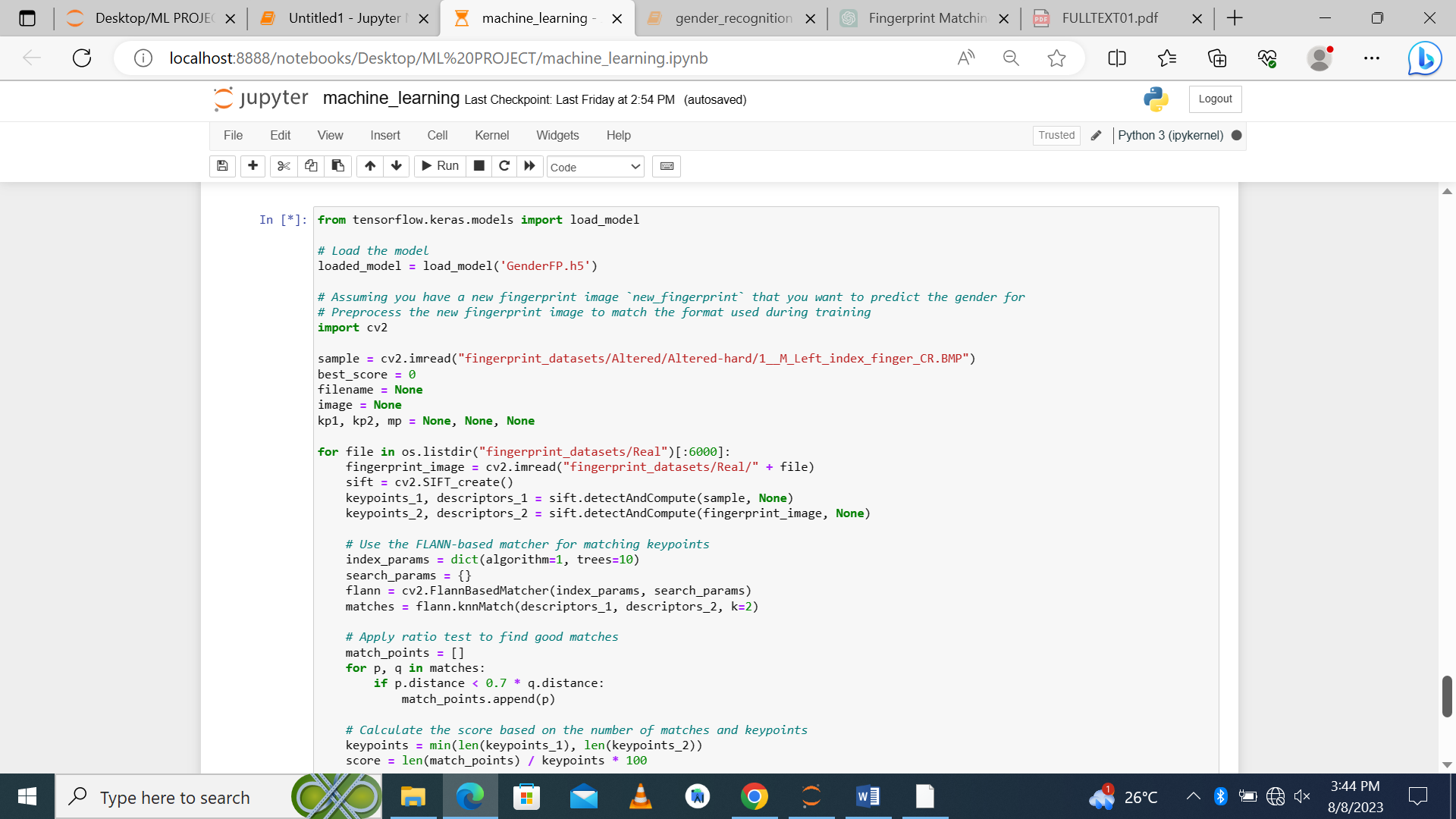
****

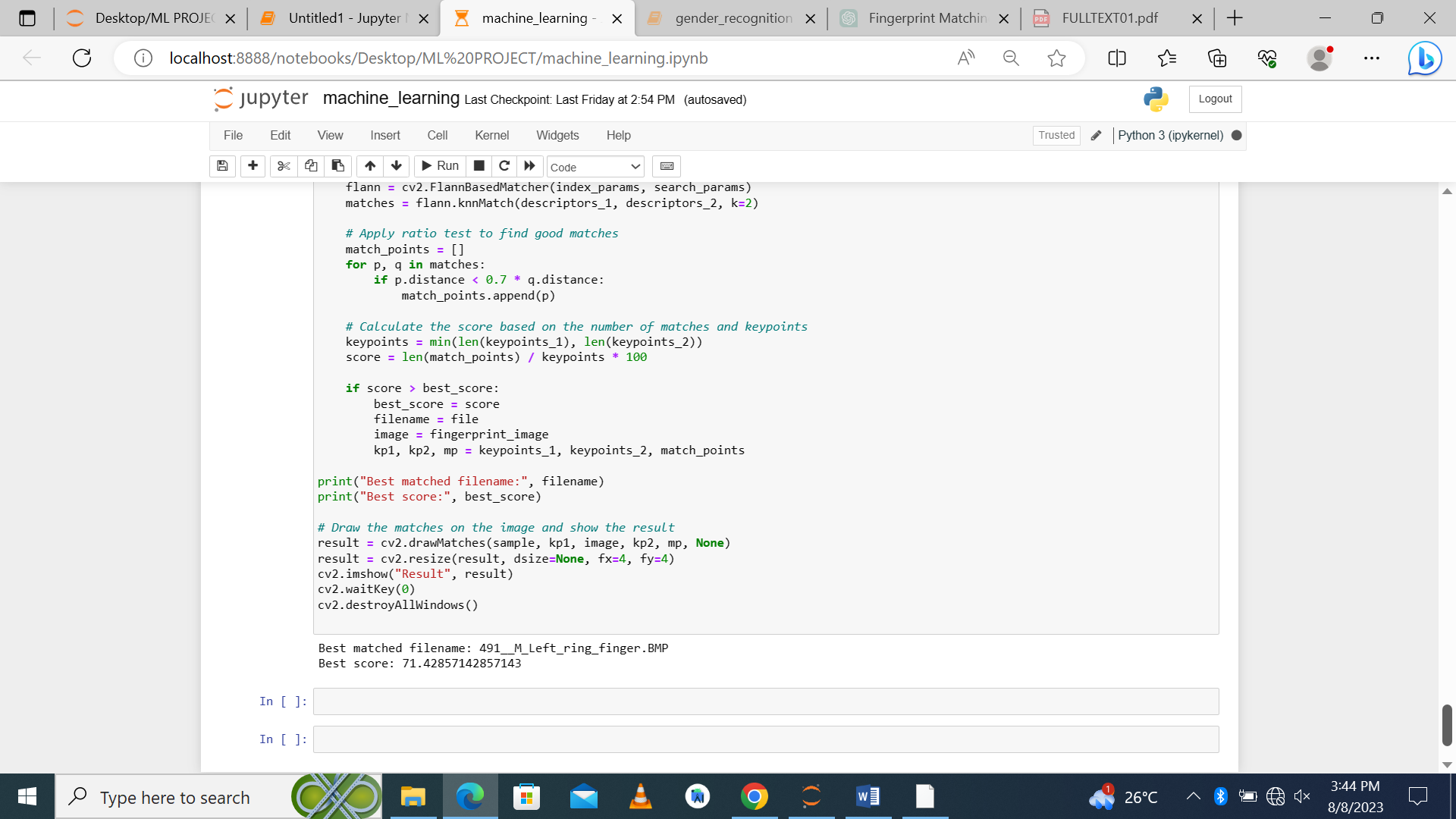
**Save the model:** Save the entire model, including the architecture, optimizer state, and learned weights, into a single HDF5 file, format is a popular file format for storing large datasets.

****

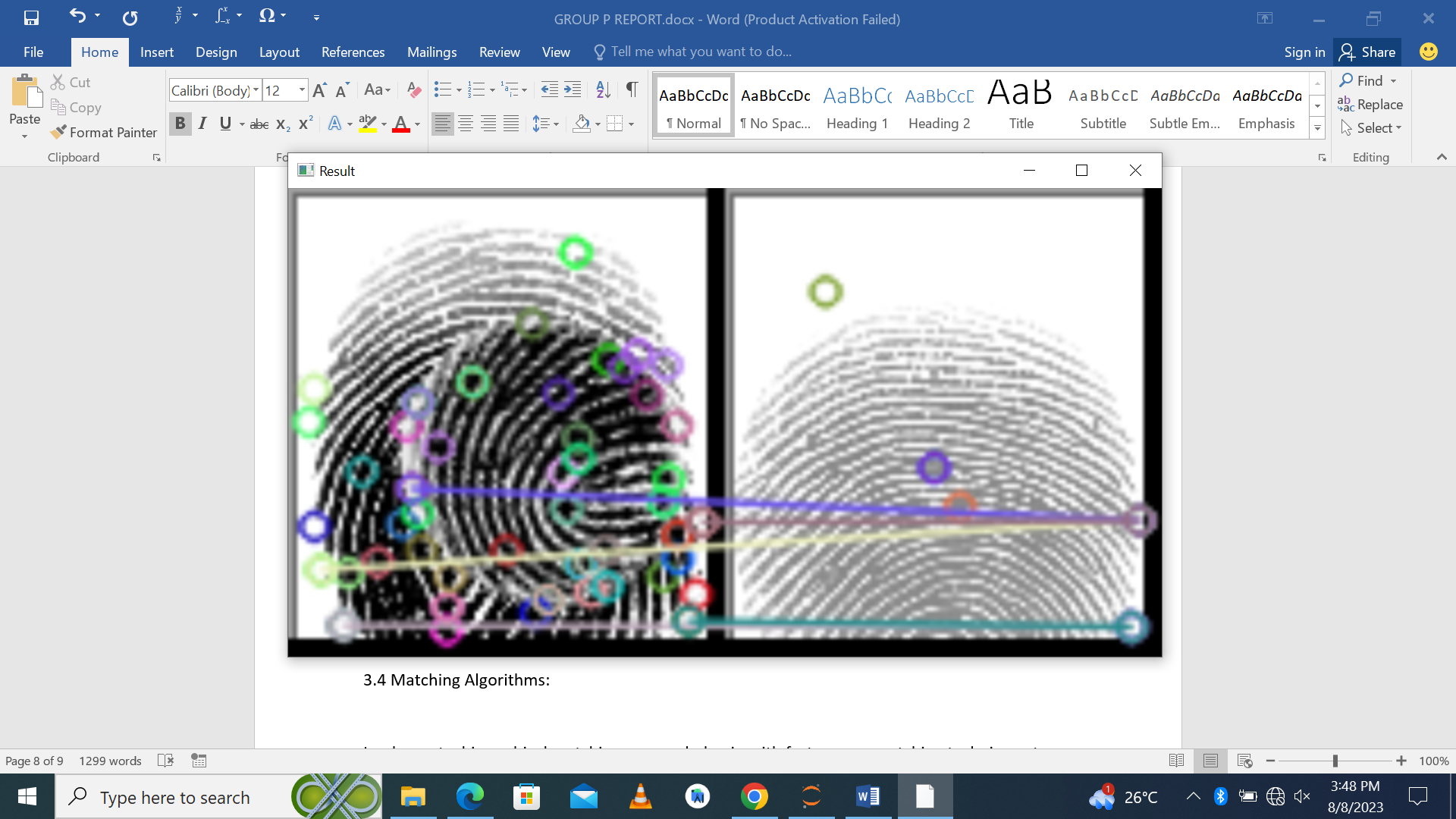
**Using the saved model to match fingerprints in other subfolders:**

After saving the model, it can then be loaded and be used for predicting and matching fingerprints in the rest of the dataset using match points and key points

****



# **Expected Outcomes:**

****

Improved accuracy and robustness in fingerprint matching, enabling reliable identification even in challenging conditions.

Reduced processing time due to the combination of fast initial matching techniques and advanced verification methods.

# **Conclusion**

This concept paper proposed a methodology for fingerprint matching that combines advanced techniques in data preprocessing, feature extraction, matching algorithms, and template protection. The outlined approach aims to address the challenges associated with fingerprint recognition and improve accuracy, efficiency, and security in real-world applications. Further research and development in this direction could lead to breakthroughs in biometric identification systems.