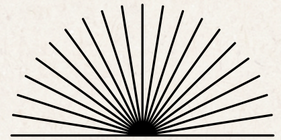
MUSIC INSTRUMENT RECOGNITION USING MACHINE LEARNING

PRESENTED BY:
DIYA DINEEP - CB.SC.U4AIE24157
VEMURI MONISHA REDDY - CB.SC.U4AIE24157



Introduction

- We use machine learning to automatically identify musical instruments in audio recordings.
- It enhances music transcription, sound classification, and automatic tagging for streaming services.
- We are using Support Vector Machines (SVM) to classify instruments using extracted audio features.
- It is chosen for its efficiency in handling high-dimensional data and strong generalization even with limited training samples.

Objectives

 # 1

To develop an SVM-based model for musical instrument recognition in monophonic recordings.

 # 2

To train and test the model on monophonic instrument datasets and evaluate performance using accuracy, precision, recall and F1-score.

 # 3

To improve classification accuracy by optimizing SVM kernel functions and hyperparameters.

Methodology

Step 1: Data Collection

Introduction to the Dataset:

The Philharmonia Dataset is a publicly available collection of musical instrument recordings provided by the Philharmonia Orchestra. It contains high-quality, monophonic samples played by professional musicians.

Structure:

- *Number of Instruments: 20*
- *Instrument Families: The dataset covers four major families of musical instruments:*

1. *Strings: Violin, Viola, Cello, Double Bass, Harp, Guitar*
2. *Woodwinds: Flute, Oboe, Clarinet, Bassoon, Saxophone*
3. *Brass: Trumpet, Trombone, French Horn, Tuba*
4. *Percussion: Timpani, Xylophone, Marimba, Snare Drum*

Step 2: Preprocessing the Audio

1. Convert MP3 to WAV

- *Since MP3 is compressed, we need to convert it into WAV .*
- *Use FFmpeg to convert MP3 to WAV.*

2.Normalize Volume & Remove Silence

- *If some recordings are too loud and some too soft, we normalize them to the same volume level.*
- *We also remove extra silence at the beginning and end of the audio files.*

3.Convert Audio to Mono

- *Some recordings in the Philharmonia dataset are in stereo, but instrument classification works best with mono audio.*
- *Converting stereo to mono simplifies processing and reduces unnecessary complexity.*

4.Convert to a Uniform Sample Rate

- *Sample rate is how many times per second the computer captures the sound.*
- *A common sample rate is 16,000 Hz or 22,050 Hz.*
- *We resample all audio to the same rate so they can be compared fairly.*

Step 3: Feature Extraction

MFCC (Mel Frequency Cepstral Coefficients):

- *Captures the shape of the sound, like how our ears perceive different frequencies.*
- *Helps in recognizing different instrument tones.*
- *Often used in speech and music recognition because it mimics human hearing.*

Spectrograms:

- *Converts sound into a visual representation (like a heatmap).*
- *Shows how different frequencies change over time.*
- *Helps in spotting patterns in musical instruments.*

Zero-Crossing Rate (ZCR):

- *Measures how often the sound wave crosses the zero line.*
- *Helps in distinguishing between percussive (drums) and melodic (flute, violin) instruments.*

Spectral Centroid:

- *Finds the center of mass of the sound's frequencies.*
- *If the energy is in high frequencies, the sound is sharp. If it's in low frequencies, the sound is deep and warm.*
- *Helps in separating bright instruments (like cymbals) from soft ones (like a bass guitar).*

Chroma Features:

- *Identifies the musical notes being played.*
- *Focuses on pitch and harmony rather than just raw sound.*
- *Helps in recognizing instruments that play different notes.*

Step 4: Train the SVM Model

Preparing the Dataset:

Before training, we need to split our dataset into two parts:

- *Training Set (80%) – This is used to teach the model how different instruments sound.*
- *Testing Set (20%) – This is used to evaluate if the model can correctly identify instruments.*

Training the SVM Model:

- *SVM takes in extracted features (MFCC, Spectrograms, etc.) and learns patterns that differentiate instruments.*
- *Types of SVM Kernels (Different Ways to Separate the Sounds):*

1. *RBF Kernel (Radial Basis Function) → Draws a curved boundary (better for complex instrument sounds).*
2. *Polynomial Kernel → Uses higher-degree curves to separate very similar sounds.*
3. *Linear Kernel → Draws a straight line to separate instruments .*

Testing and Selecting the Best Model:

- *After training, we test the model using the 20% test data.*
- *The model predicts the instrument, and we compare it with the actual labels.*
- *The accuracy tells us how well the model performs.*

Learning to Separate Instrument Sounds

- *Each instrument has unique features (MFCC, Spectrogram, Chroma, etc.).*
- *SVM learns to separate different instruments based on these extracted features.*

Finding the Best Boundary

- *SVM finds a boundary that keeps the maximum distance between instrument categories.*
- *Support Vectors (important points) define the boundary.*

Handling Complex Data

- *If the sounds overlap, a straight line won't work.*
- *Kernel Trick helps transform data into a higher dimension for better separation.*
- *Types of Kernels Used:*

✓ *Linear Kernel – For simple separation*
✓ *RBF Kernel – Best for complex musical sounds*
✓ *Polynomial Kernel – If feature relationships are polynomial-like*

Making Predictions

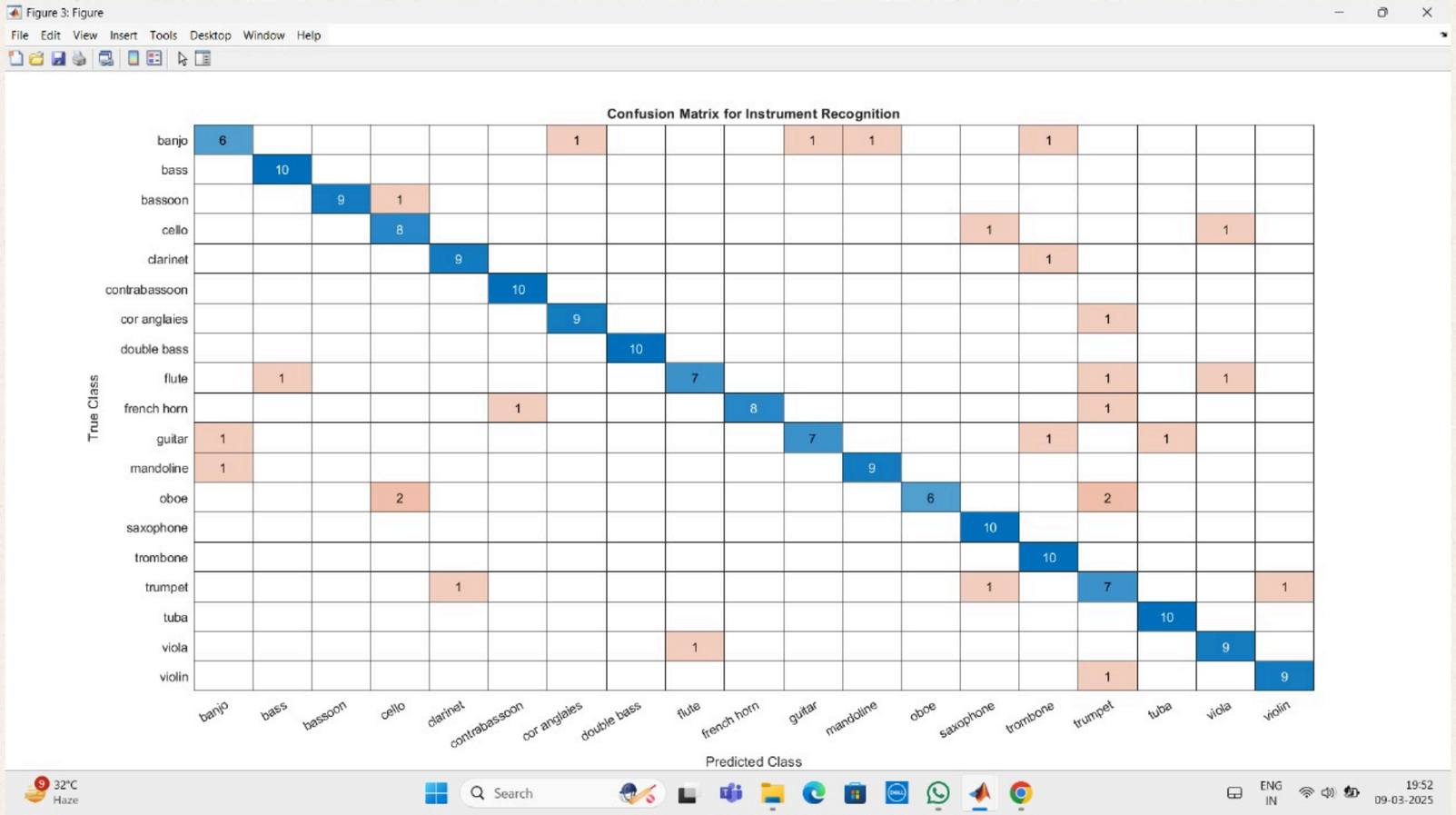
- *After training, SVM predicts the instrument for a new sound.*
- *It checks which side of the boundary the new sound falls on and classifies it correctly.*

Step 5: Prediction

- Load a new audio file (a recording of an unknown instrument).
- Extract features (MFCC, spectrogram, etc.) from this new file.
- Feed these features into the trained SVM model.
- The model predicts which instrument it is.

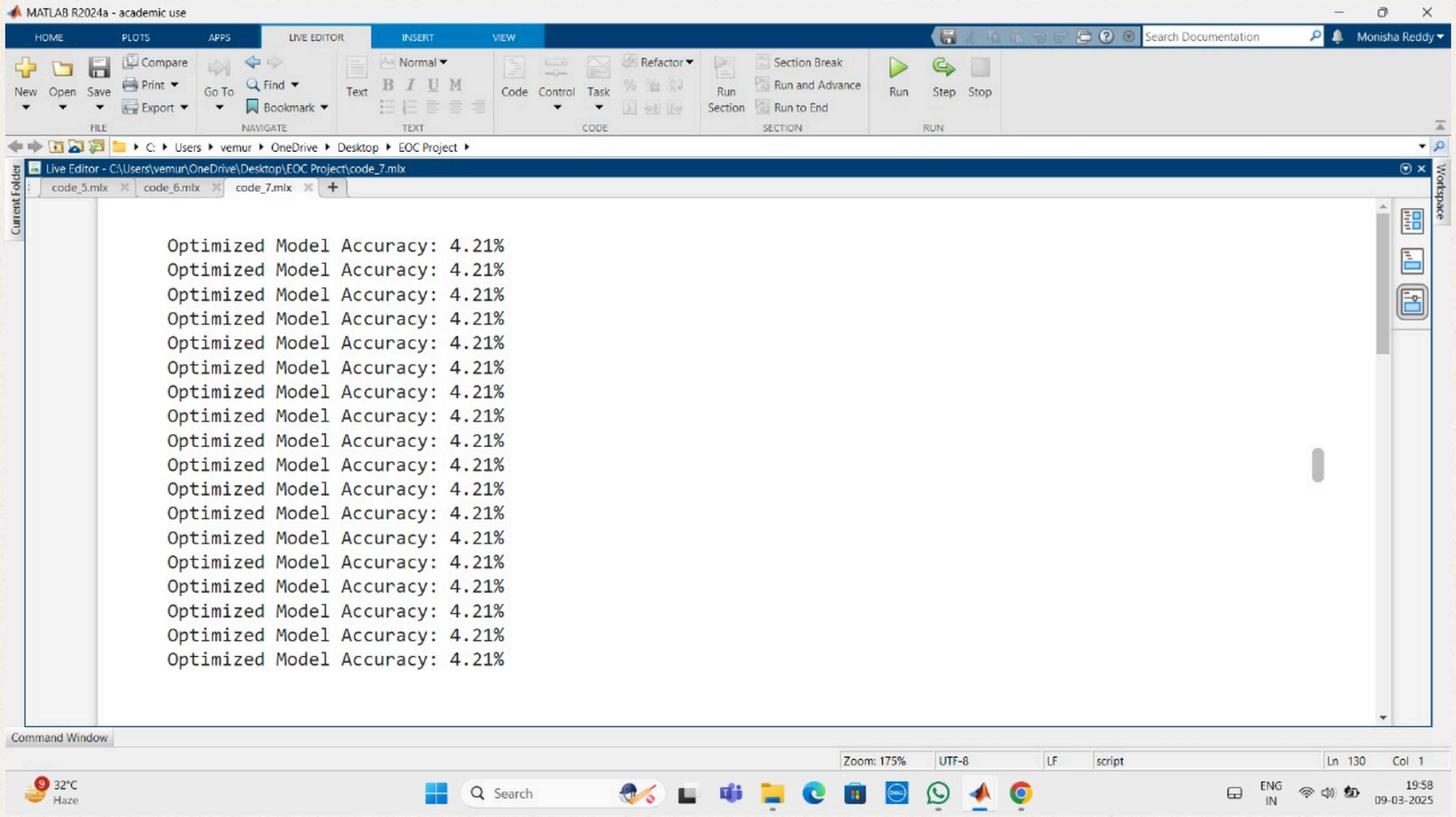
Results

Confusion Matrix



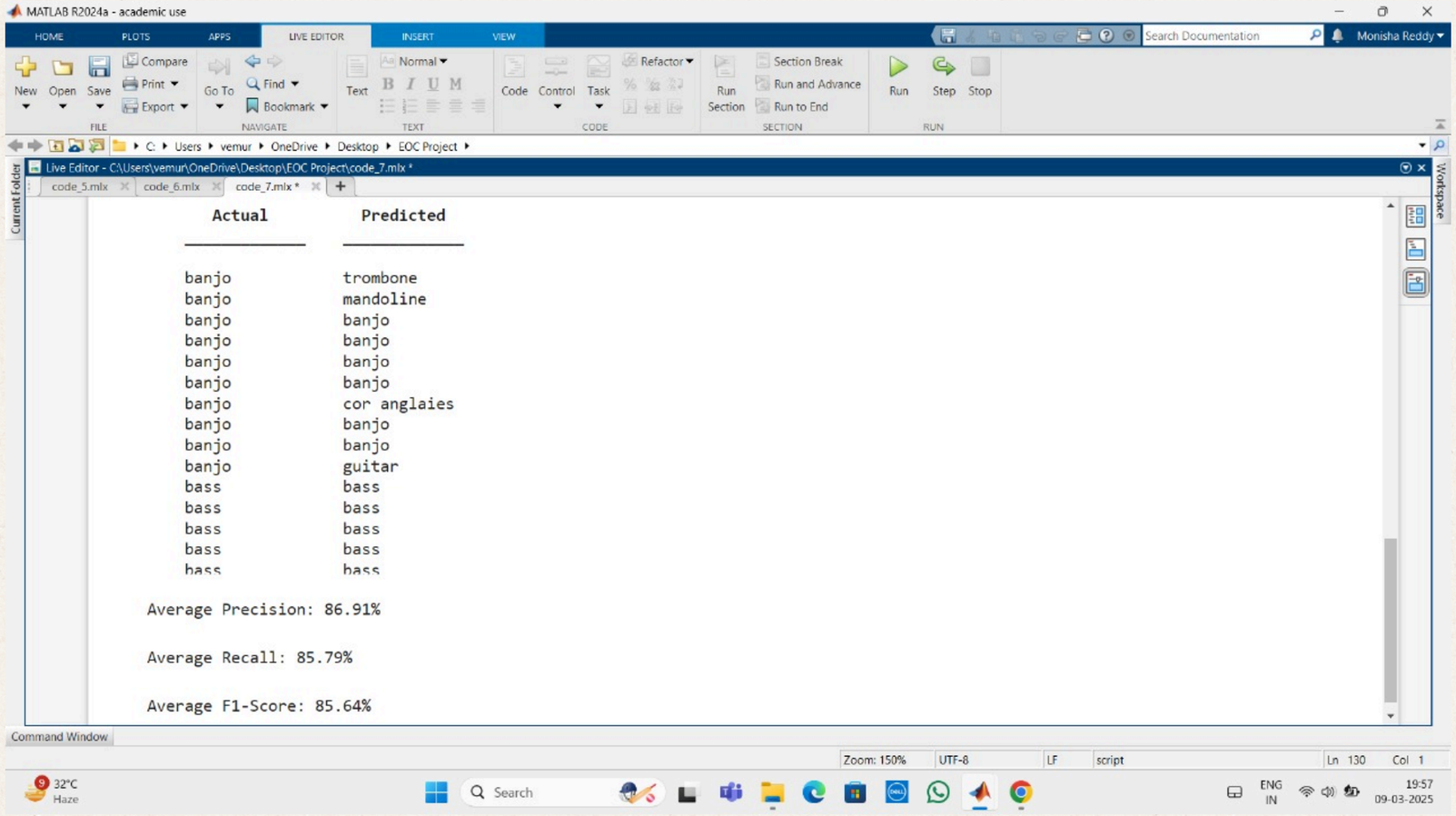
Results

Accuracy



Results

Precision, Recall, F1 Score



THANK YOU!