Comparison of Performance of AI models on Audio MNIST Dataset

- Ajay K. Basu
akbasu48@gmail.com
(Single member team)
Aug. 27, 2025

This project has been taken up for fulfilment of requirements of the Stanford University Continuing Study Course of Summer 2025 - **Applied Machine Learning with Python**
Course Code**:** TECH 27 Instructor Name**: Ishaani Priyadarshini**

1. **Problem Statement:** Comparison of Performance of AI models on Audio MNIST Dataset for classifiaction of digits (0 - 9)

2. **Objectives and goals:**

Objective:

To study performance of various AI models on the same dataset with same or similar parameters and hyper parameters to compare performance in terms of common metrics (as applicable) and explain the results.

a. Training and Validation accuracy
b. Loss
c. Confusion Matrix
d. Classification Report on Precision, Recall on F1-score
e. Mean Square Error (MSE)
f.
g. Training Time required to achieve convergence / accuracy

Goal:

This study is expected to give us deep understanding on choice of model and parameters for solving a given problem (in this case a classification problem) optimally.

3. **Methodology:**

a. Platform:
Jupyter Notebook on Sky Tech Server with NVIDIA GPU is used as the main platform.

b Tools:

Python tools -

tensorflow, keras on tensorflow and sklearn have been used for model building, training and evaluation; matplotlib has been used for plotting.
selenium and shutil for development of video

4. Dataset: Audio MNIST

About Dataset:

Audio MNIST dataset from keras has been used for modelling. This data set available to us has 30,000 data files in the following format:

a. Main folder: Audio MNIST
b. Subfolders: 60 subfolders (00 to 59), one for each speaker, who contributed their voice.
c. Files: each subfolder has 500 files in the format digit_speaker_sample.wav
digit: 0 - 9; speaker: 00 - 59 samples: 50 per digit : 00 - 49

Hence the file 5_43_21 indicates digit =5, speaker_id = 43, sample_id = 21

Size: Each file is about 70 KB; total 3,000 files; total dataset size is about 1.91 GB

About data:

a. Format: `.wav` (16-bit PCM, mono)

b. Sampling Rate: 48 kH

c. bit: 16

d. channels: 1 (mono)

e. Content: recordings of spoken digits (0–9)

f. Speakers: 60 speakers (male + female)

g. Utterances: each speaker repeats each digit 50 times

h. Duration per file: ~1–2 seconds

File size:

Estimated file size (KB) = Sample Rate in Khz × Bit Depth × Channels × Duration (1- 2 sec) / 8

= 48 x 16 x 1 x 1 to 48 x 16 x 1 x 2 KB

= 76.8 - 153.6 KB

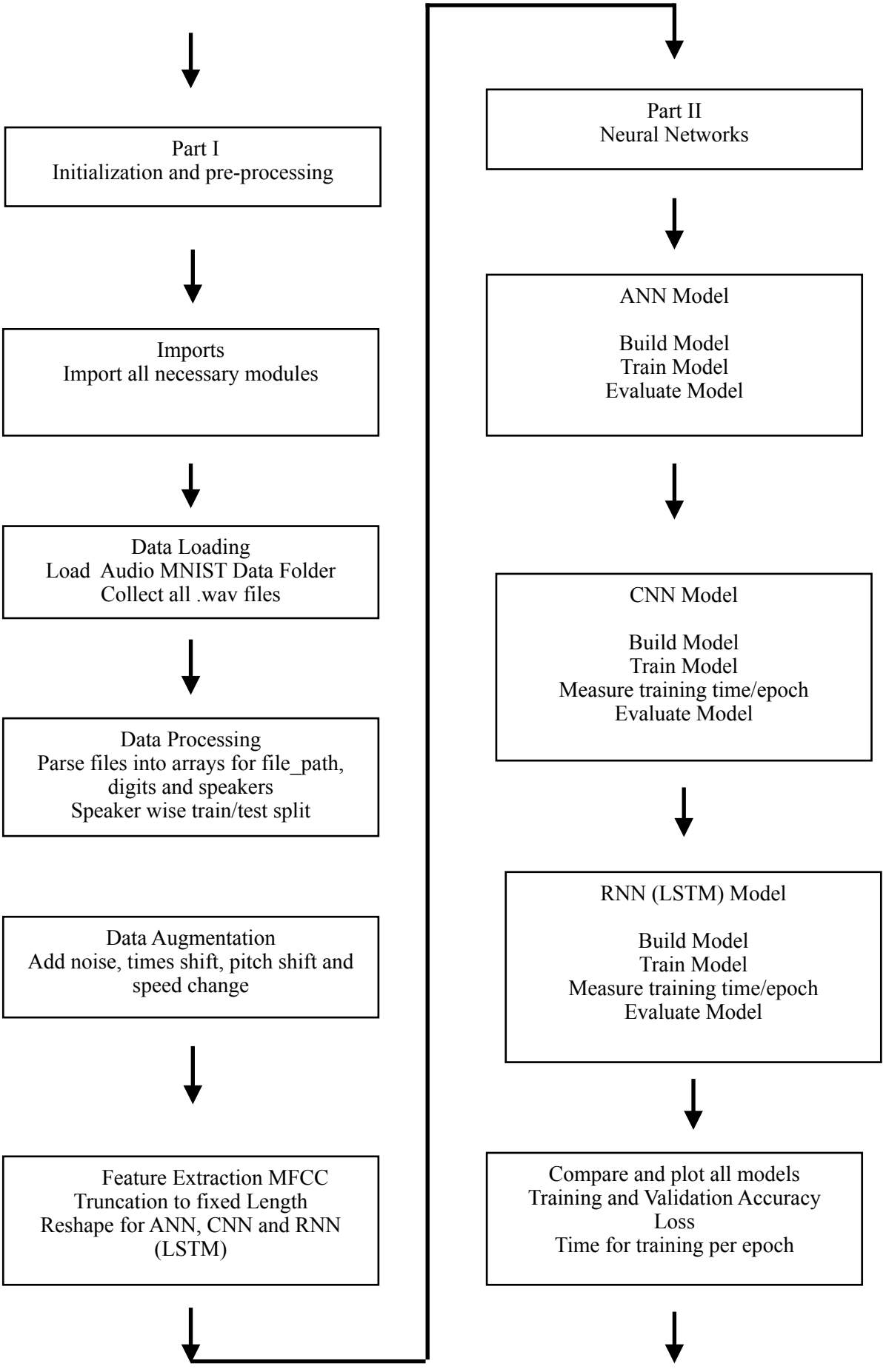
Practically sizes start from around 70 KB indicating that the speakers spoke faster than estimated.

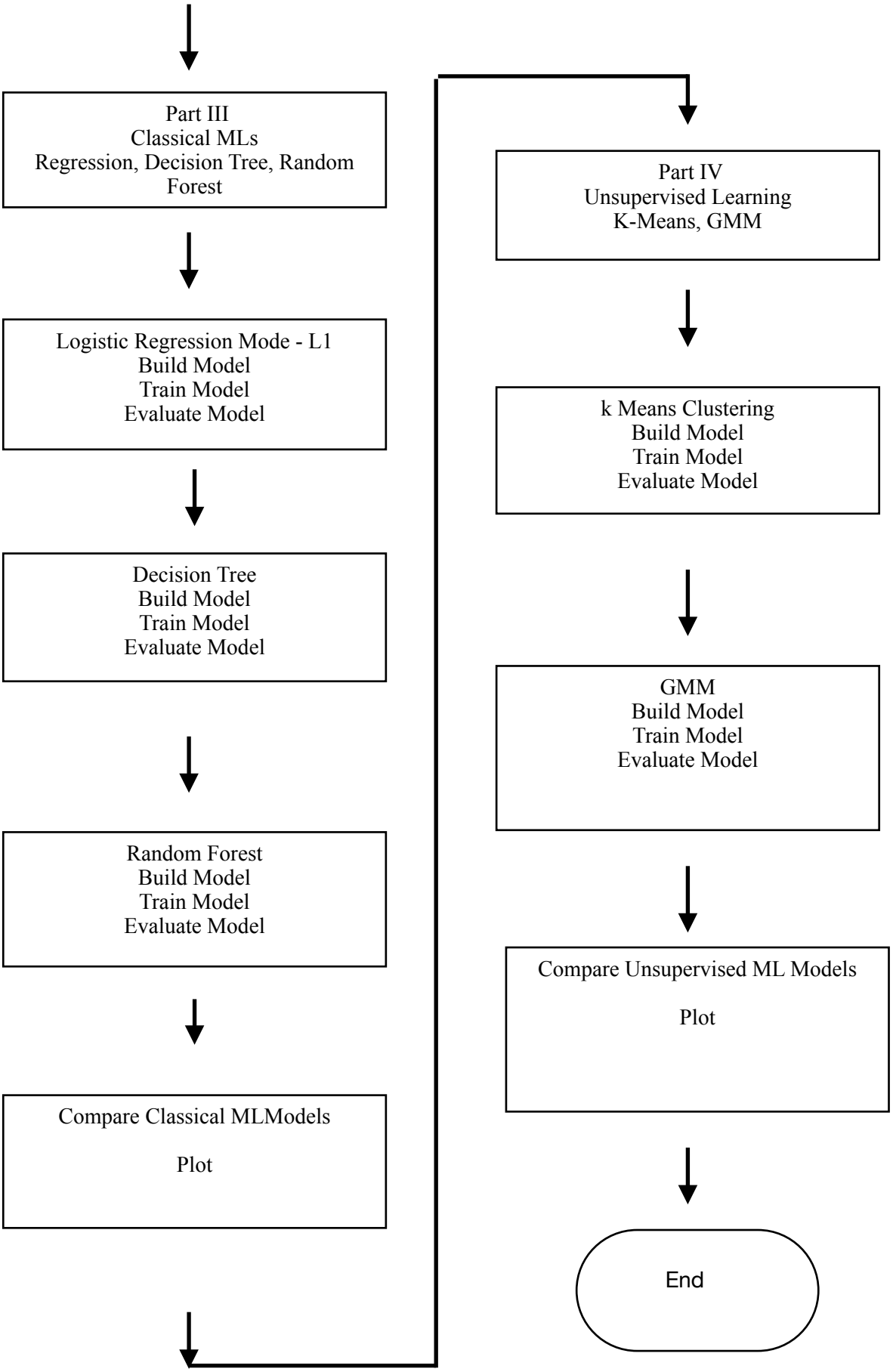
Why choose this dataset?

a. I wanted to experiment with audio data

c. It is effectively a time series - ideal for experimenting with CNN, RNN (LSTM) or Transformers.

5. Block Diagram:




Part III
Classical MLs
Regression, Decision Tree, Random Forest
Logistic Regression Mode - L1
Build Model
Train Model
Evaluate Model
Decision Tree
Build Model
Train Model
Evaluate Model
Random Forest
Build Model
Train Model
Evaluate Model
Compare Classical MLModels
Plot
Part IV
Unsupervised Learning
K-Means, GMM
k Means Clustering
Build Model
Train Model
Evaluate Model
GMM
Build Model
Train Model
Evaluate Model
Compare Unsupervised ML Models
Plot
End

6. Data Visualization - Exploration and Preprocessing:

i. All the audio (.wav) file were joined together in 'all files'.

ii. Parsed and converted to numpy arrays for file path, digits and speakers

iii. Sample checked for accuracy.

iv. Speaker wise data splitting:

Data is split into train and test data in 90:10 ratio speaker wise to make sure same speaker is not used for both training and testing.

v. Data Augmentation:

Data augmentation is necessary for audio data as Audio signals change with environment (background noise, echo, room acoustics), speakers (pitch, accent, speed), and devices (microphone quality). A model trained on clean data may fail when tested on noisy or slightly different recording. Augmentation simulates these conditions, making the model robust with improved model invariance (model need to focus on content (in this case spoken digits), not irrelevant variations.

Small dataset problem - Collecting large labeled audio datasets (speech, environmental sounds, music) is time-consuming and expensive.

In this project, we have added the following to improve the model:

Noise - background sound
Time shift - speech spoken earlier or later
Pitch change - male vs female voice
Speed change - slower vs faster

7. Feature Engineering:

Feature engineering is needed to turn messy waveforms into meaningful, compact, and discriminative features.

Techniques: time-domain (ZCR, energy), frequency-domain (spectrograms), cepstral (MFCC, LPC).

MFCCs are the most widely used audio features, especially for **speech** and **speaker recognition** and **has been chosen for the following reasons:**

Perceptual Relevance:

Uses **Mel scale** (nonlinear frequency scale aligned with human ear perception). Humans perceive pitch logarithmically, and MFCC mimics that.

Compact Representation:

From a full spectrogram with thousands of values per frame → MFCC compresses into ~13–40 coefficients. In this case 40 coefficients have been chosen.

Reduces dimensionality: faster training.

Noise Robustness:

Captures spectral shape rather than raw amplitudes; more robust to background noise and microphone variations.

8. Machine Learning Models Used:

Three Neural Network models - ANN, CNN and RNN

Three classical ML models - Logistic Regression (L1), Decision Trees and Random Forest,

Two unsupervised learning models - K-Means and GMM

9. Experiments:

Experiments have been done with parameters and hyper parameters, particularly with Unsupervised Learning Models as the results shows no clear distinguishable clusters. More experiments necessary before concluding that Unsupervised Learning is unsuitable for this type of dataset (audio problems)

10. Evaluation Metrics Chosen:

i. Neural Networks :

Accuracy: Accuracy = Correct Predictions / Total Predictions
Good for balanced datasets, misleading for imbalanced ones.

Loss : Log Loss (Cross-Entropy Loss)
Measures the uncertainty of predictions.

ii. Classical ML models:

Precision: Out of predicted positives, how many are correct
TP / TP + FP

Recall (Sensitivity): Out of actual positives, how many were predicted correctly.
TP / TP + FN

F1 Score: Harmonic mean of Precision and Recall -

2 x Precision X Recall / Precision + Recall

11. Comparison Tables and Graphs (Summary):

Neural Network Models: After 50 epochs

Training

Model	Batch Size	Training time (per epoch in secs)	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
ANN	32	5.5	0.9918	0.0575	0.9493	0.7922
CNN	32	2.0	0.9971	0.0125	0.9647	0.3089
RNN	32	1.0	0.9966	0.0153	0.9583	0.2724

Evaluation

Model	Accuracy	Loss
ANN	0.9493	0.8275
CNN	0.9647	0.3089
RNN	0.9583	0.2704

Remarks:

ANN (Feedforward Network):

Very high training accuracy (99.2%) but
validation loss is much higher (0.79). Indicates overfitting - the model memorizes training data but generalizes less effectively.
Validation accuracy (94.9%) is good, but worse than CNN/RNN.CNN

CNN(Convolutional Neural Network):

Best overall accuracy (96.5%) and relatively low validation/evaluation loss (0.31). Captures spatial/temporal patterns in MFCC spectrogram-like inputs effectively.

Slight overfitting (gap between training loss 0.0125 vs. validation loss 0.31) but still generalizes very well.

RNN (Recurrent Neural Network):

Training accuracy ~99.7%, close to CNN. Validation accuracy (95.8%) slightly lower than CNN, but validation loss (0.27) is even better than CNN.Suggests RNN captures temporal dependencies in audio sequences effectively.

Balanced performance: less overfitting than ANN, slightly better loss profile than CNN.

Classical ML Models:

Accuracy and Error

Model	Average Test Accuracy	Error MSE
Log Reg L2	0.9200	1.1797
Decision Tree	0.7900	3.9950
Random Forest	0.9510	0.6873

Classification Report (Average)

Model	Pecision	Recall	F1-Score
Log Reg L2			
Decision Tree	0.7900	0.7900	0.7900
Random Forest	0.9500	0.9500	0.9500

Confusion Matrix

Decision Tree

218	5	29	11	6	1	2	26	0	2
3	229	5	3	9	17	0	6	3	25
12	4	213	19	3	12	12	18	5	2
6	4	17	242	4	3	6	2	6	10
11	2	16	5	221	21	5	8	3	8
3	12	2	4	11	243	9	9	0	7
8	0	4	10	1	3	262	7	4	1
8	4	18	22	5	12	3	222	0	6
0	0	8	8	1	0	1	0	280	2
7	14	2	13	2	20	2	6	7	227

Random Forest

275	0	13	5	0	1	0	6	0	0
0	289	1	1	2	5	0	0	0	2
15	0	276	6	0	0	1	2	0	0
1	2	8	284	0	2	0	2	1	0
0	4	1	0	281	10	0	2	1	1
0	0	0	0	0	297	0	3	0	0
0	0	1	3	0	0	295	0	1	0
0	0	6	10	0	1	1	282	0	0
0	0	1	6	0	0	1	1	291	0
0	6	1	2	0	0	0	2	6	28

Remarks:

The strong Random Forest result shows that ensemble methods are well-suited for the AudioMNIST task, balancing complexity with generalization.

Logistic Regression has a competitive linear baseline, showing that MFCC features already carry useful information.

Decision Tree, while interpretable, is too unstable for high-dimensional speech data.

Unsupervised Models:

Model	Silhouette Score	Adjusted Rand Index
K-Means	0.0296	0.0036
GMM	-0.0688	0.0009

Remarks :

Unsupervised model scores are near zero. Clusters are overlapping or not well separated. Data points are almost equally close to multiple clusters.

12. Optimization Results:

Model	Accuracy in %	No. of epochs	Average training time per eopoch in seconds
ANN	98	12	5.5
CNN	99	9	2.0
RNN	99	25	1.0

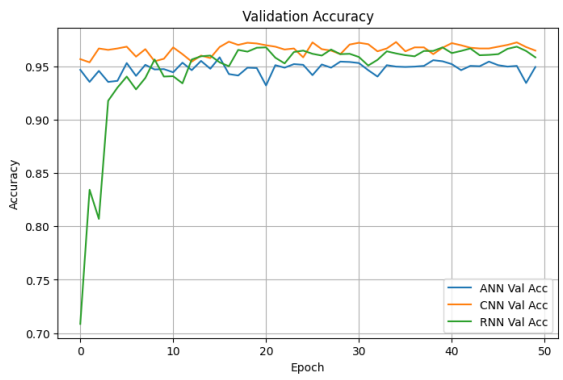
Note : Training was continued for 50 epochs for comparison

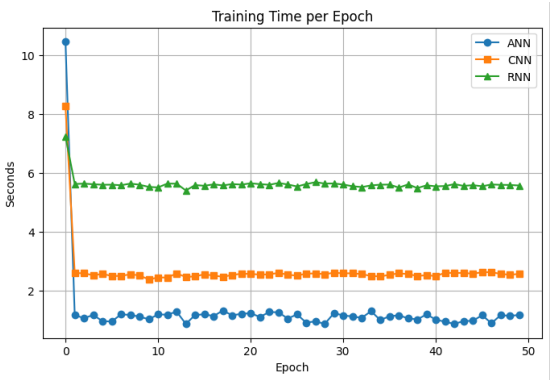
MFCC: = 40

PCA : Original shape: (30000, 4000) Reduced shape: (30000, 1000)

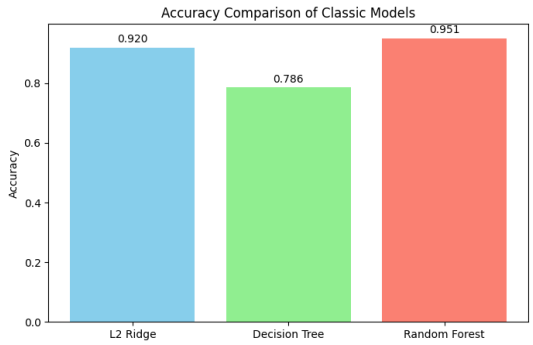
13. Testing Graphs:

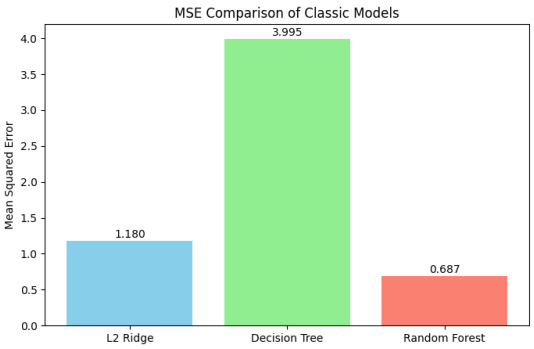
Neural Networks





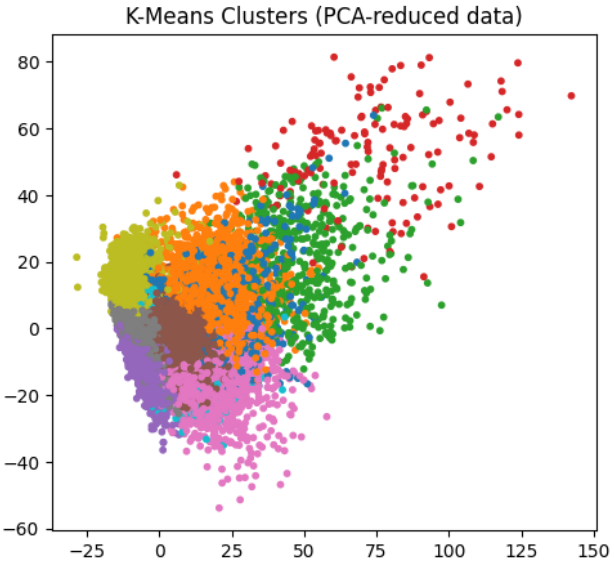
Classic ML Models

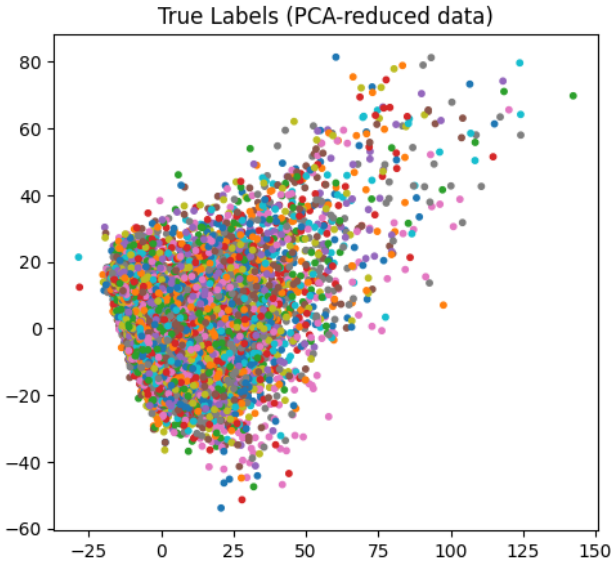




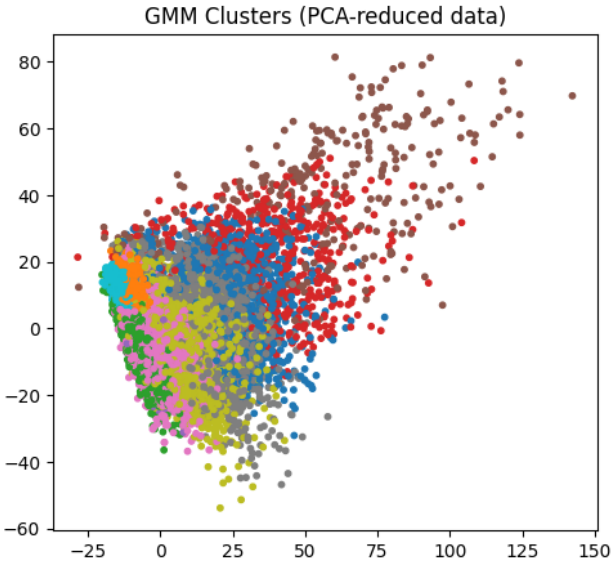
Unsupervised Models

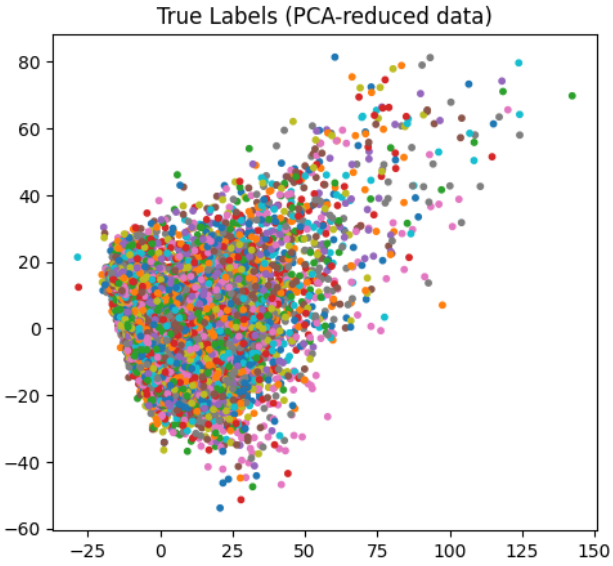
K-Means



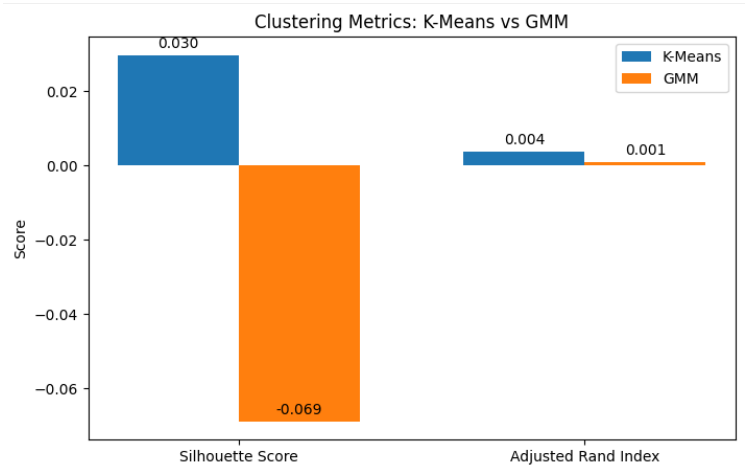


GMM





Comparison Plot (K-means vs GMM)



14. Standards and Constraints:

a. Major constraints were resource (availability of strong server).

b. Google Colab was initially chosen as platform but abandoned as the computation time allotted in a session was insufficient.

15. Bias, Ethics and Fairness - No issues

16. Limitations - Processing Power

17. Video or Demo: The ipynb notebook has been converted to a 1-minute mp4 video which has been tested on quicktime < notebook_video.mp4>.

18. Conclusion & Future Woks:

Feature extraction is extremely important for this task. Dynamic features like ΔMFCC, ΔΔMFCC extraction may capture temporal changes better. Spectrogram / log-Mel spectrogram instead of MFCCs can be tried. Speaker normalization may also improve results.

For Neural Networks, **Both CNN and RNN produced good results**. A **CNN-RNN** hybrid model, where CNN extracts local spectral features and RNN layer models temporal features may improve performance further.

For the **classical ML models are not suitable for high dimensional and non-linear data**, but still tried for comparison. **Logistic Regression** works well only when classes are linearly separable in feature space, which does not hold true in this case. **Decision Tree** has a poor result, possibly because of overfitting. **Random Forest** is balanced and has the best result but not easily interpretable.

Unsupervised learning models produced worst results. Models appear to be clustering something else than the digits. Possible reason could be **K-Means** assume spherical clusters and **GMM** assumes Gaussian Distribution, which may not hold for audio data. **Spectral Clustering** or **Deep Embedding Clustering (DEC) models** may be tried out.

19. Acknowledgement:

Prof. Ishaani Priyadarshini and Stanford University for introducing so much in a such a short time with examples, reading materials, assignments and quizzes.

20. Suggestions: It is suggested that the course be expanded to a **16**-session course.