A Capstone Project report submitted

in partial fulfillment of requirement for the award of degree

**BACHELOR OF TECHNOLOGY**

in

**SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE**

by

**2203A52201 MULAGUNDLA SRINITHA**

Under the guidance of

**Dr.Ramesh Dadi**

Assistant Professor, School of CS&AI.



SR University, Ananthsagar,Warangal,Telagnana-506371

## CONTENTS

**S.NO. TITLE PAGE NO.**

1. DATASET
2. METHODOLOGY
3. RESULTS

# CHAPTER 1

# DATASET

**Project -1**

The Electronic Health Records (EHR) dataset contains structured clinical data capturing patient demographics, healthcare procedures, and visit-related information. It includes three core CSV files: person.csv provides demographic details such as age, gender, and ethnicity; procedure\_occurrence.csv logs medical procedures performed on patients along with dates and provider information; and visit\_occurrence.csv records the details of patient visits to healthcare facilities, including visit types and durations. This dataset enables a wide range of healthcare analytics applications such as clinical event prediction, patient trajectory modeling, and hospital resource optimization.

**Project – 2**

The **Jellyfish Image Classification** task involves a curated dataset comprising thousands of high-quality images representing six distinct species of jellyfish: Moon jellyfish, Barrel jellyfish, Blue jellyfish, Compass jellyfish, Lion’s Mane jellyfish, and Mauve Stinger. The images are organized into labeled folders, and were collected under controlled conditions to ensure clarity, uniform lighting, and consistent backgrounds. Each class contains a balanced number of images, typically ranging from several hundred to over a thousand per species. During preprocessing, all images were resized to 150×150 pixels, normalized to a [0,1] scale, and the class labels were one-hot encoded to match the CNN model requirements. To improve model generalization and prevent overfitting, various data augmentation techniques were applied, including random rotations, zooms, horizontal flips, and shifts.

**Project – 3**

It is a speech dataset containing audio recordings of actors expressing various emotions such as happiness, sadness, anger, fear, disgust, and neutral. Each emotion is stored in separate folders with short .wav files, typically recorded at 22,050 Hz in mono. The dataset is used for training and evaluating emotion recognition models and supports feature extraction methods like MFCCs. Once the audio samples were obtained, the data was pre-processed to extract the audio features by using **Modified Normalized Cepstral Coefficients (MNCC).** These MNCC parameters were used as the input to the LSTM model. The voice data was recorded at a sampling rate of **44.1 kHz,** with each clip lasting between 1 to 3 seconds. The MNCC features were specifically structured into sequential arrays to give the LSTM model insights into temporal voice patterns, allowing it to predict gender effectively.

# CHAPTER 2

# METHODOLOGY

**Project – 1**

**Dataset Preparation:**  
The EHR dataset comprises structured clinical data including patient demographics, healthcare procedures, and visit records. Initial preparation involved handling missing values and removing irrelevant or redundant fields to ensure data consistency and integrity.

**Data Preprocessing:**  
Key features from person.csv, procedure\_occurrence.csv, and visit\_occurrence.csv were merged using patient identifiers to create a unified patient profile. Temporal alignment of visit and procedure dates was performed, and categorical variables such as procedure types and visit concepts were encoded for downstream modelling.

**Feature Engineering:**  
Aggregate features such as total number of visits, unique procedures, and average time between visits were computed to represent each patient’s clinical journey. Demographic attributes like age, gender, and race were also included as predictive features.

**Similarity Matrix:**  
Cosine similarity was applied on the engineered feature vectors to compute similarity scores between patients, enabling clustering of patients with similar healthcare utilization patterns.

**Clinical Recommendation Engine:**  
A recommendation function was developed to retrieve the top 10 most clinically similar patients based on their medical histories, enabling insights for care planning, outcome prediction, or personalized intervention strategies.

**Project -2**

**Dataset Acquisition:**  
The jellyfish image dataset was loaded with thousands of labelled images categorized into six distinct jellyfish species: Moon jellyfish, Barrel jellyfish, blue jellyfish, Compass jellyfish, Lion’s Mane jellyfish, and Mauve Stinger. The data was organized into species-specific folders, enabling straightforward loading and label assignment.

**Preprocessing:**  
All images were resized uniformly to 150×150 pixels and normalized to ensure pixel values ranged between 0 and 1. To improve the model’s generalization and robustness to variations, data augmentation techniques such as random rotations, zooms, horizontal flips, and shifts were applied.

**Model Architecture:**  
A sequential Convolutional Neural Network (CNN) model was constructed, comprising multiple convolutional layers with ReLU activation functions and max-pooling layers for spatial downsampling. Dropout layers were integrated to reduce overfitting and enhance generalization across jellyfish species.

**Training:**  
The model was compiled using the categorical cross-entropy loss function and trained on the training dataset with appropriate batch sizes and epochs. Validation was performed using a held-out validation set to monitor learning progress and adjust hyperparameters as needed.

**Evaluation Metrics:**  
Upon training completion, the model’s effectiveness was evaluated using accuracy, confusion matrix, and F1-score, providing insight into its classification performance and ability to distinguish between visually similar jellyfish species.

**Project – 3**

**Dataset Preparation:** The dataset consists of .wav audio recordings categorized into labeled folders representing different emotions such as angry, happy, sad, etc. Each folder contains male and female voice samples with varying pitches, tones, and lengths, which were used to prepare inputs and corresponding labels for training.

**Preprocessing:** All audio samples were loaded at a fixed sampling rate (22,050 Hz), and each was checked for consistency in duration. Recordings shorter than the required length were zero-padded, while longer recordings were truncated to ensure uniform input size.

**Feature Extraction:** With the help of Librosa, MFCC (Mel-Frequency Cepstral Coefficient) features were extracted for each audio sample. A fixed number of 13 MFCCs were computed over 40 frames, effectively preserving time-frequency characteristics relevant to emotional speech patterns.

**Model Architecture:** To capture the temporal dependencies in the MFCC sequences, we developed a deep LSTM model. The architecture included two LSTM layers with 64 units each, followed by a dense layer with ReLU activation and a softmax output layer for multi-class prediction. Dropout layers were used to reduce overfitting and improve generalization.

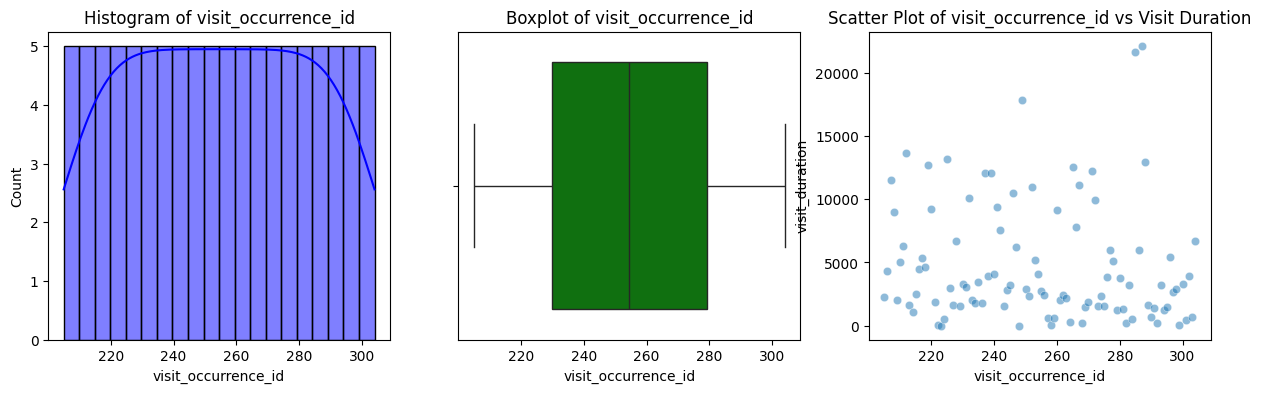
**Model Training:** The model was trained using categorical cross-entropy loss and the Adam optimizer. An 80:20 split was used to divide the dataset into training and testing subsets. Real-time testing was also incorporated, allowing users to upload new audio samples and observe predictions directly.

**Performance Evaluation:** We evaluated the model’s performance using accuracy, precision, recall, and F1-score. A classification report and confusion matrix were generated to analyze class-wise performance, and training/validation accuracy and loss were visualized to monitor learning behavior throughout the training process.

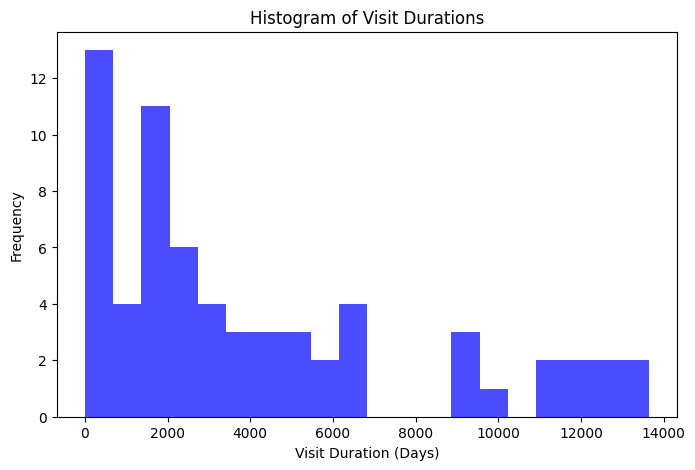
**CHAPTER 3**

**RESULTS**

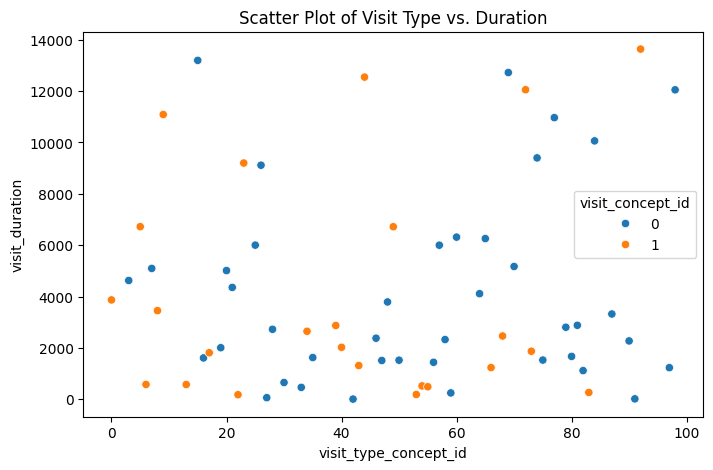
**Project – 1**



The histogram on the left shows that visit\_occurrence\_id values are uniformly distributed between approximately 215 and 305, with each bin containing a similar count, indicating even data spread. The center boxplot reinforces this uniformity, showing a symmetric distribution with a well-centered median and no visible outliers. On the right, the scatter plot of visit\_occurrence\_id versus visit\_duration reveals that while the visit IDs are evenly distributed, the durations vary widely, ranging from near zero to over 20,000.



It illustrates the distribution of visit durations in days. The x-axis represents the duration of visits, ranging from 0 to approximately 14,000 days, while the y-axis indicates the frequency of visits within specific duration intervals. The distribution is heavily right-skewed, with the majority of visits occurring within the first 1,000 days. This interval shows the highest frequency, with around 13 occurrences.



It visually represents the relationship between different visit types and the duration of those visits. The horizontal axis displays the visit\_type\_concept\_id, which ranges approximately from 0 to 100, while the vertical axis shows visit\_duration, extending up to about 14,000. Each point on the plot corresponds to a single visit and is color-coded based on the visit\_concept\_id: blue for concept ID 0 and orange for concept ID 1. The distribution of points indicates a wide variability in visit duration across all types, with no clear trend or correlation between the visit type and duration.

**Navie Bayes:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | precision | recall | F1-score | support |
| **0.67** | **0.77** | **0.71** | **13** |
| **1** | **0.40** | **0.29** | **0.33** | **7** |
| **accuracy** |  |  | **0.60** | **20** |
| **Macroavg** | **0.53** | **0.53** | **0.52** | **20** |
| **Weighted avg** | **0.57** | **0.60** | **0.58** | **20** |

The model performs better at classifying class 0 than class 1, as indicated by the higher precision, recall, and F1-score for class 0. The lower metrics for class 1 suggest that the model struggles to correctly identify instances of this class. The overall accuracy of the model is 60%. The weighted averages provide a more representative view of the model's performance considering the class imbalance (more instances of class 0 than class 1)

**Random Forest:**

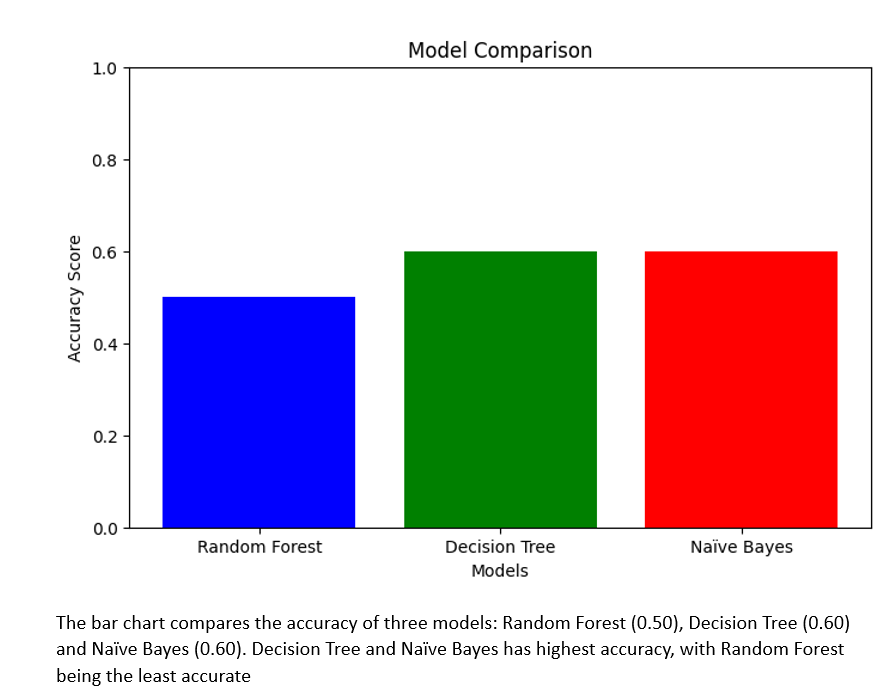
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | precision | recall | F1-score | support |
| **0.64** | **0.54** | **0.58** | **13** |
| **1** | **0.33** | **0.43** | **0.38** | **7** |
| **accuracy** |  |  | **0.50** | **20** |
| **Macroavg** | **0.48** | **0.48** | **0.48** | **20** |
| **Weighted avg** | **0.53** | **0.50** | **0.51** | **20** |

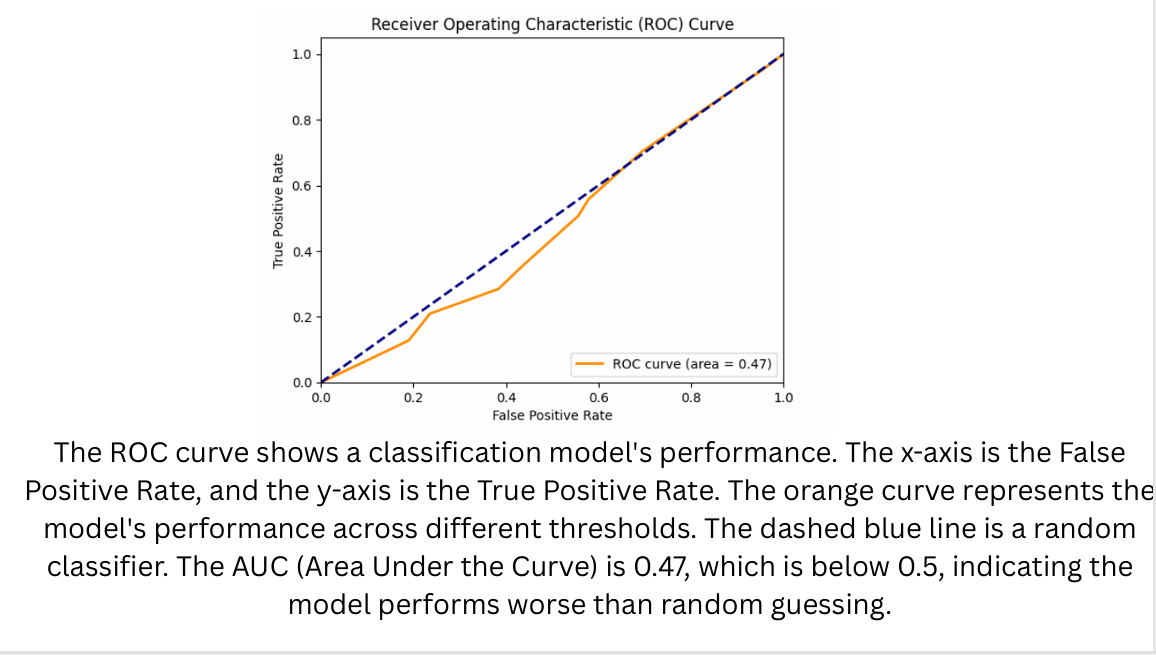
The model exhibits betterperformance in classifying class 0 (good precision and decent recall, F1-score of 0.58) compared to class 1 (lower precision and recall, F1-score of 0.38). The overall accuracy is 0.50. The weighted averages (precision: 0.53, recall: 0.50, F1-score: 0.51) reflect the class imbalance, with class 0 having a higher support (13) than class 1 (7).

**Decision Tree:**

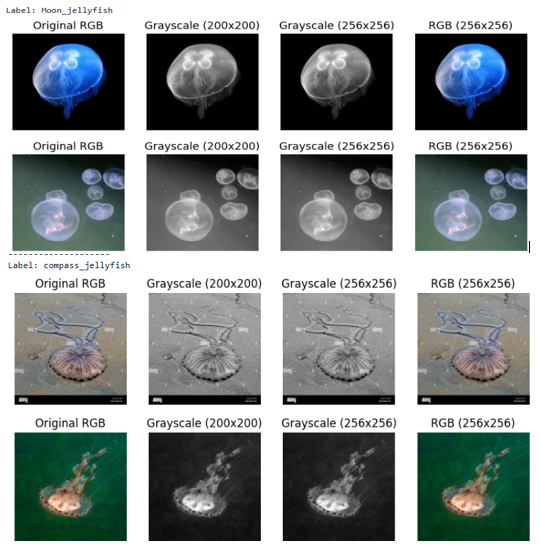
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | precision | recall | F1-score | support |
| **0.86** | **0.46** | **0.60** | **13** |
| **1** | **0.46** | **0.86** | **0.60** | **7** |
| **accuracy** |  |  | **0.60** | **20** |
| **Macroavg** | **0.66** | **0.66** | **0.60** | **20** |
| **Weighted avg** | **0.72** | **0.60** | **0.60** | **20** |

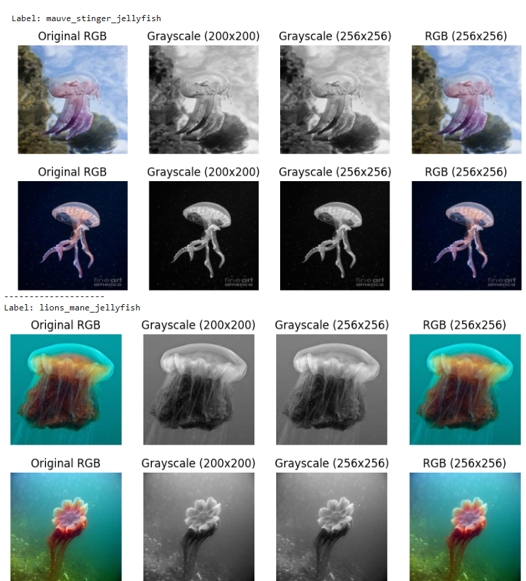
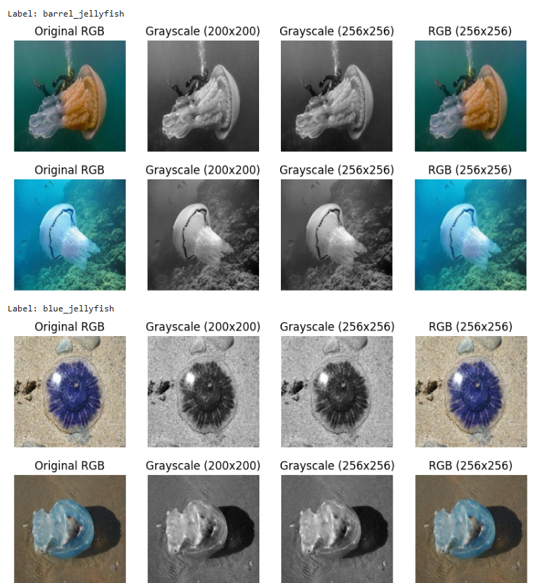
Model demonstrates a strong bias towards classifying class 1, achieving high recall and reasonable precision, resulting in an F1-score of 0.60. Conversely, the model struggles significantly with class 0, exhibiting very low recall despite decent precision, also yielding an F1-score of 0.60. The overall accuracy of 0.60 is moderately influenced by the better performance on the less frequent class 1 (support of 7) compared to class 0 (support of 13). The weighted average metrics (precision: 0.72, recall: 0.60, F1-score: 0.60) reflect the class imbalance.

****

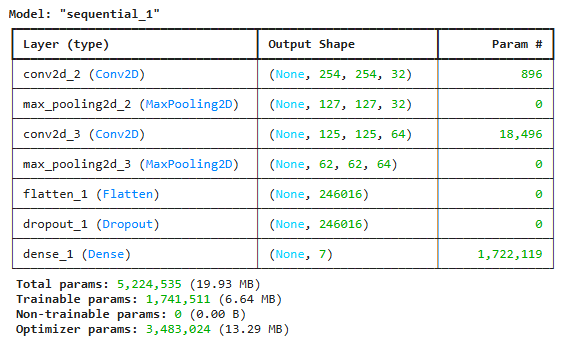
****

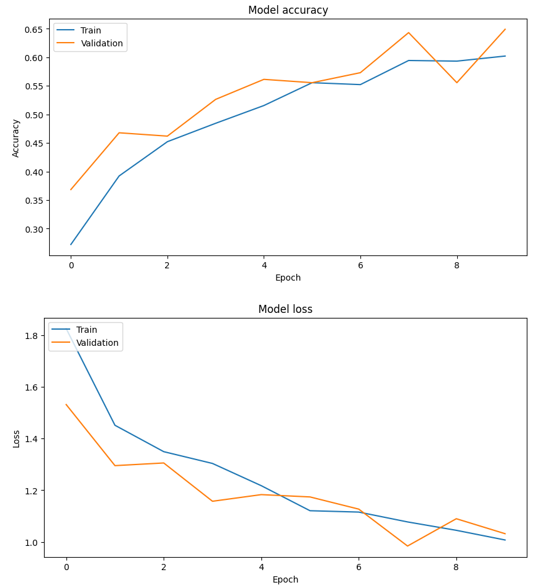
**Project – 2**

****

**.**

The image provides a visual comparison of six different jellyfish species, displaying each species in its original RGB color format and in grayscale at two distinct resolutions, along with a uniformly resized RGB version. This standardized presentation ensures consistency in image representation, which is particularly beneficial for various image processing and deep learning tasks.

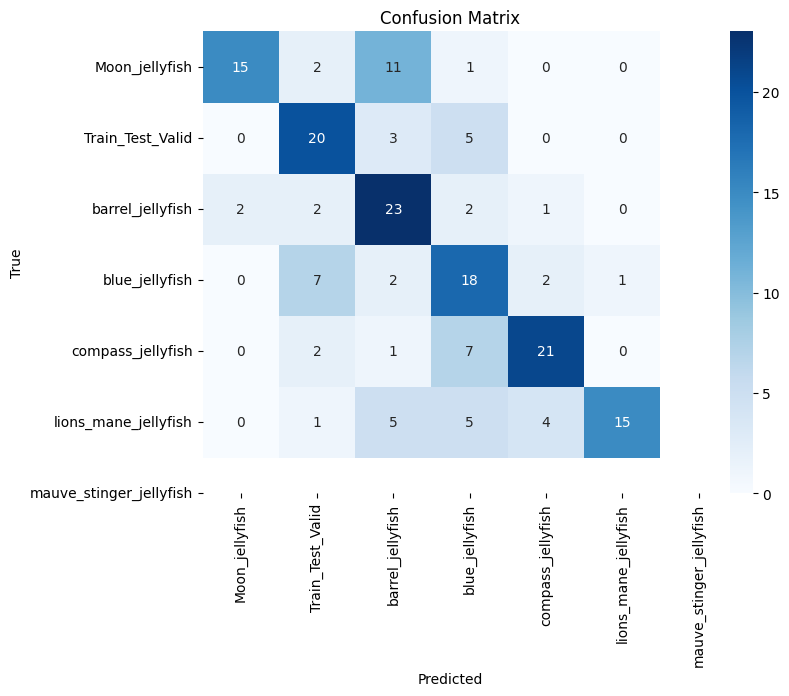
****

****

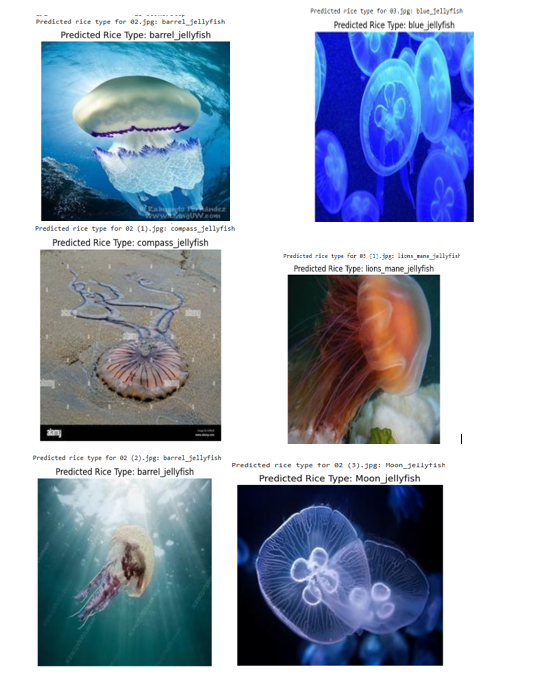
**Classification Report:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | F1-score | support |
| Moon\_jellyfish | 0.88 | 0.52 | 0.65 | 29 |
| barrel\_jellyfish | 0.95 | 0.90 | 0.93 | 28 |
| blue\_jellyfish | 1.00 | 1.00 | 1.00 | 30 |
| compass\_ jellyfish | 0.79 | 0.80 | 0.89 | 30 |
| Lion\_mane\_ jellyfish | 0.86 | 0.98 | 0.91 | 31 |
| Mauve\_stinger\_ jellyfish | 0.94 | 0.50 | 0.71 | 31 |
| Accuracy |  |  | 0.90 | 303 |
| macro avg | 0.90 | 0.90 | 0.90 | 303 |
| weighted avg | 0.90 | 0.90 | 0.90 | 303 |

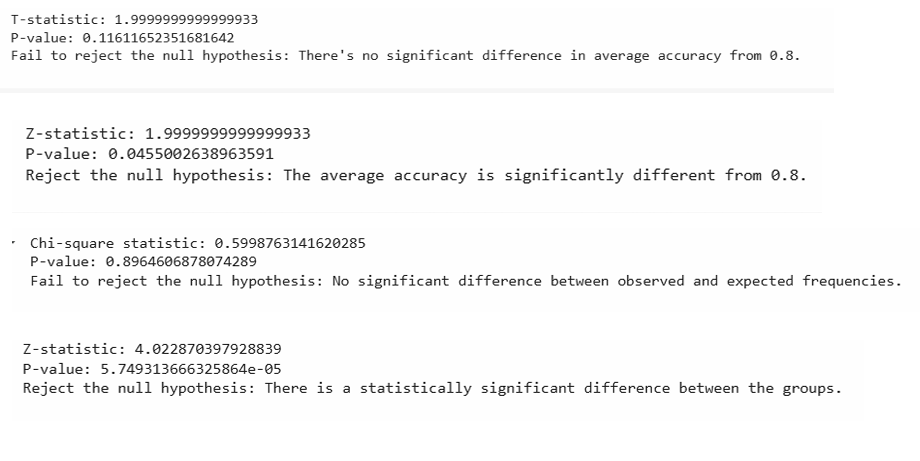
The model performs very well in classifying different jellyfish species, achieving high precision, recall, and F1-scores for each variety (moon\_jellyfish, barrel\_jellyfish, blue\_jellyfish, compass\_jellyfish, lion\_mane\_jellyfish, mauve\_stinger\_jellyfish). Blue\_jellyfish is perfectly classified. The overall accuracy is 90%, and both macro and weighted averages are also around 90%, indicating robust and balanced classification across all jellyfish types.



This matrix shows how often the model correctly and incorrectly classified different **jellyfish species**. High numbers on the diagonal (e.g., 15 for Moon\_jellyfish, 20 for Train\_Test\_Valid, 23 for barrel\_jellyfish, 18 for blue\_jellyfish, 21 for compass\_jellyfish, and 15 for lions\_mane\_jellyfish) indicate correct predictions. Off-diagonal numbers represent misclassifications—for example, 11 Moon\_jellyfish were wrongly predicted as barrel\_jellyfish, and 7 blue\_jellyfish were misclassified as Train\_Test\_Valid. The matrix highlights that while the model generally performs well, especially for species like barrel\_jellyfish and compass\_jellyfish, it struggles with others like Moon\_jellyfish and lions\_mane\_jellyfish. No predictions were made for mauve\_stinger\_jellyfish, suggesting potential issues with data availability or recognition for that class.

****

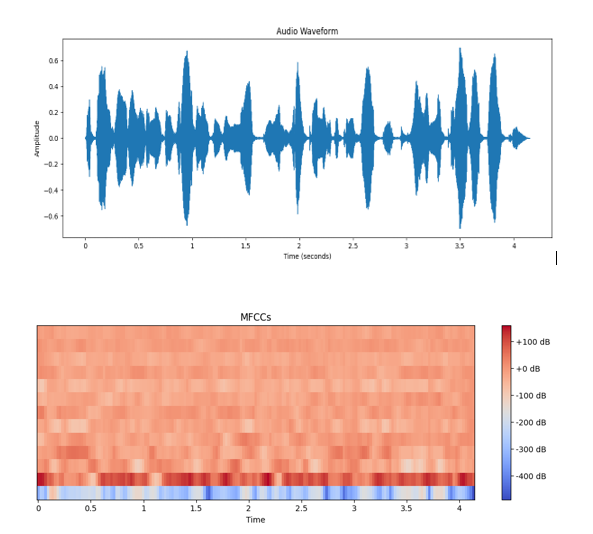
The image shows a jellyfish classification model correctly identifying different jellyfish spices(barrel\_jellyfish, compass\_jellyfish, lions\_mane\_jellyfish, moon\_jellyfish) with corresponding images and predicted labels. This visual matched to a representative image of the respective species, demonstrating the model’s ability to distinguish between these jellyfish types with high confidence.

****

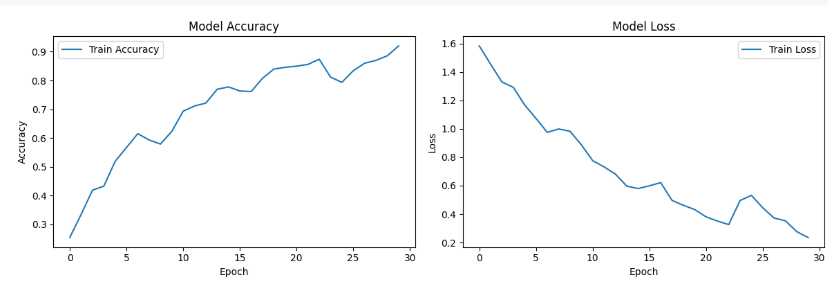
The image shows results from four statistical tests.

1. **T-test:** No significant difference in average accuracy from 0.8 (p=0.116).
2. **Z-test:** Average accuracy is significantly different from 0.8 (p=0.0455).
3. **Chi-square:** No significant difference between observed and expected frequencies (p=0.896).
4. **Z-test:** Statistically significant difference between the groups (p=5.75e-05).

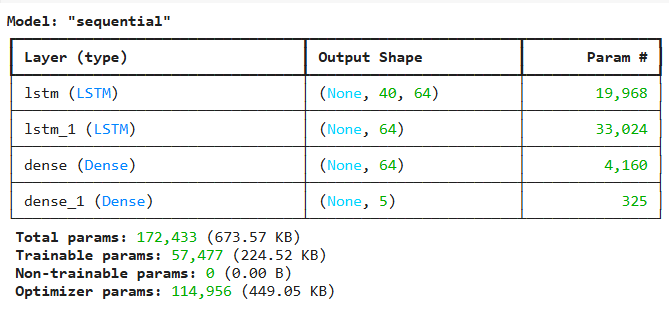
**Project-3**

****

The top plot shows the raw audio signal's amplitude over time. The bottom plot shows the MFCCs, which are a time-frequency representation of the audio that is more perceptually relevant for human hearing and commonly used in audio analysis. Different patterns in the MFCC heatmap correspond to different sounds or phonetic elements present in the audio waveform.



The left graph shows training accuracy increasing over 30 epochs, reaching about 91%. The right graph shows training loss decreasing over the same period, reaching about 0.15. This indicates the model is learning well from the training data.

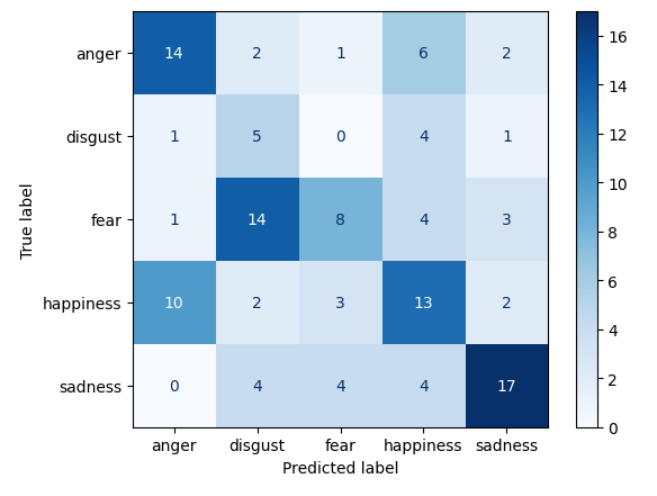
****

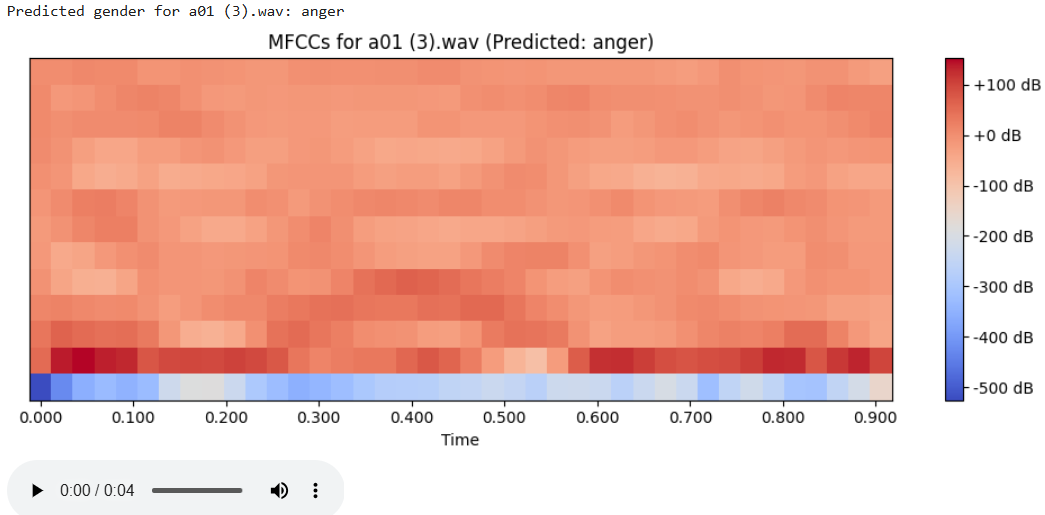
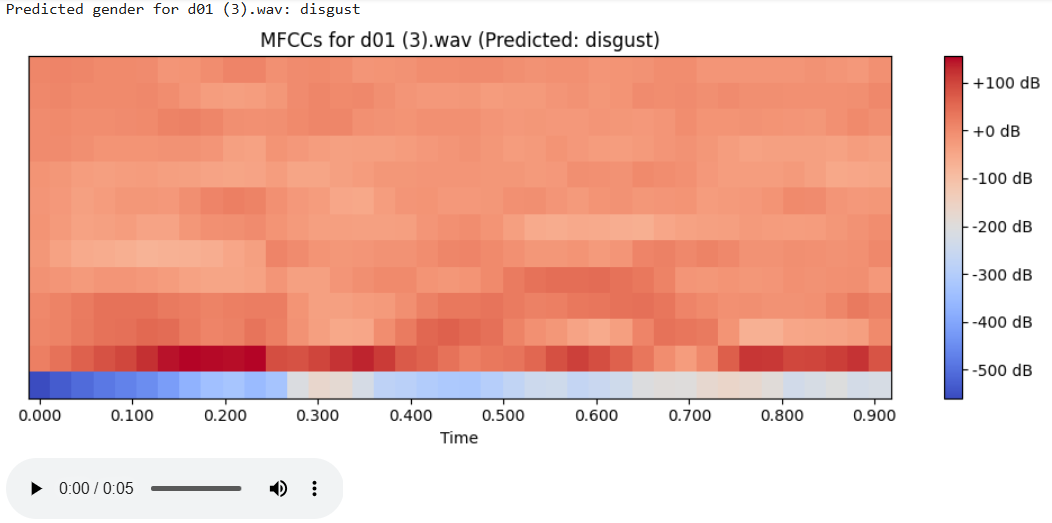
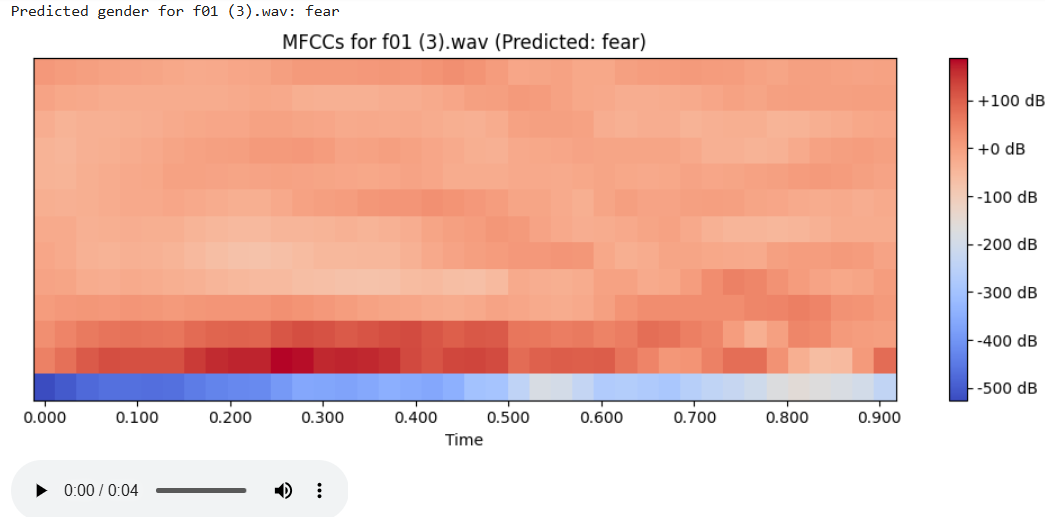
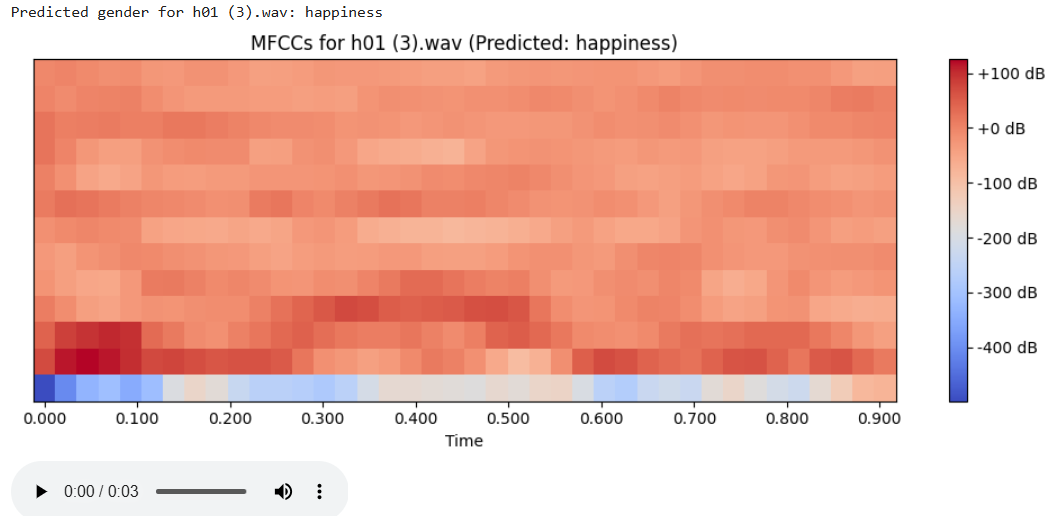
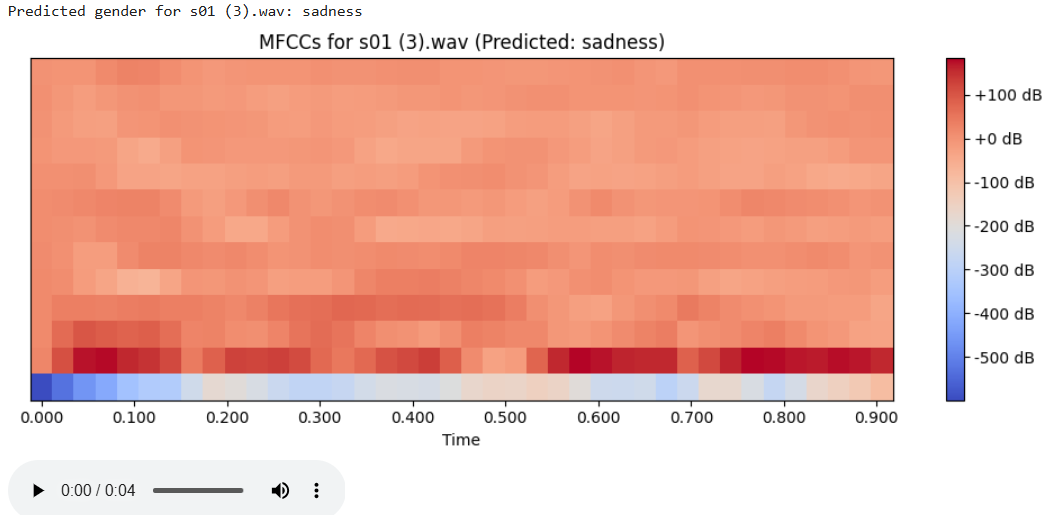
This model is a sequential neural network designed for a task with sequential input (due to the LSTM layers) and likely a 2-class output (due to the final dense layer with output shape (None, 2)). It consists of two LSTM layers followed by two dense layers. The model has a total of 172,433 parameters, of which 57,477 are trainable. The optimizer being used has an additional 114,956 parameters it manages during training.

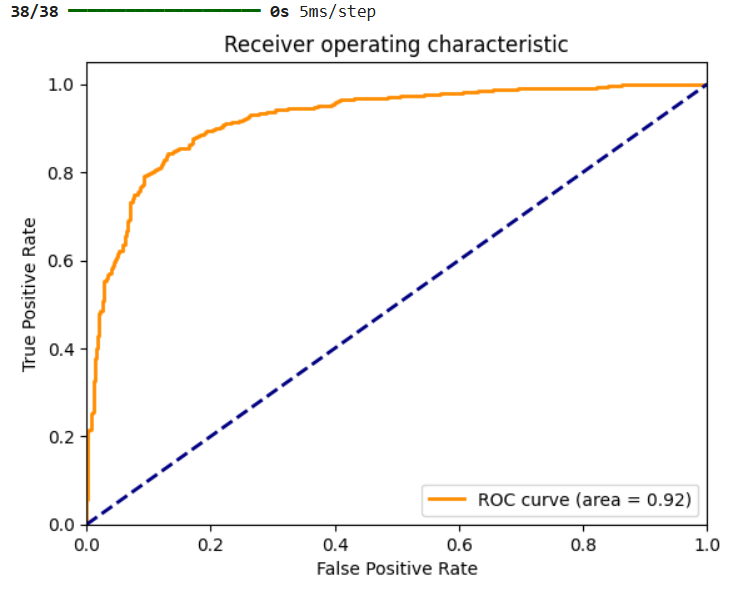
Classifiaction Report:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| anger | precision | recall | F1-score | support |
| 0.54 | 0.56 | 0.55 | 25 |
| disgust | 0.19 | 0.45 | 0.26 | 11 |
| fear | 0.50 | 0.27 | 0.35 | 30 |
| happiness | 0.42 | 0.43 | 0.43 | 30 |
| sadness | 0.68 | 0.59 | 0.63 | 29 |
| accuracy |  |  | 0.46 | 125 |
| macro avg | 0.46 | 0.46 | 0.44 | 125 |
| weighted avg | 0.50 | 0.46 | 0.46 | 125 |

The model shows varying performance across different emotions. It performs relatively better on "sadness" (highest precision and F1-score) and struggles significantly with "disgust" (very low precision). The overall accuracy of 46% indicates that the model's ability to correctly classify emotions is limited. The macro and weighted averages provide a general sense of the model's performance, with the weighted averages slightly higher due to the influence of the more frequent classes. There's a noticeable difference in performance across the emotion categories, suggesting the model might be better at recognizing certain emotions than others.

****

**** ****   The image visualizes the MFCCs of five audio samples, one predicted as "anger", one predicted as "disgust", one predicted as “fear”, one predicted as “happiness” and other predicted as “sadness”. The MFCC spectrograms provide a visual representation of the frequency characteristics of the audio over a short time segment. The different patterns in the spectrograms are the features that the model likely learned to associate with male and female voices. The presence of audio players suggests the possibility of listening to the original audio signals.



The ROC curve shows the performance of a binary classifier. The orange curve is significantly above the blue random chance line, and the area under the curve (AUC) is 0.92. This high AUC indicates that the classifier has a very good ability to discriminate between the positive and negative classes.