Project Title:

- speech emotion recognition

Project Overview:

- The project aims to develop a machine learning model capable of recognizing and categorizing emotions expressed in spoken language

Understanding emotions from speech is crucial for improving human computer interactions, enhancing user experience in various applications such as virtual assistants, customer service, and mental health monitoring. Traditional methods of emotion recognition often rely on visual cues, which are not always available or practical. This project addresses the challenge of accurately detecting emotions using only audio signals.

The data analysis involves extracting features from audio signals, such as prosody, pitch, rhythm, and spectral properties. These features are then used to train machine learning models, such as Convolutional Neural Networks (CNNs), Long Short Term Memory networks (LSTMs), or hybrid models, to classify emotions like happiness, anger, sadness, and frustration. The significance lies in the model's ability to provide real time emotion detection, which can be integrated into various applications to enhance user interaction and provide valuable insights into emotional states

Speech emotion recognition can be incredibly useful in various day to day applications. Here are some practical examples:

1. Virtual Assistants : Enhancing virtual assistants like Siri, Alexa, or Google Assistant to understand and respond to users' emotions can make interactions more natural and effective. For instance, if a user sounds frustrated, the assistant can offer more empathetic responses or additional help.

2. Customer Service: In call centers, emotion recognition can help identify when a customer is upset or angry, allowing agents to respond more appropriately and escalate calls when necessary. This can improve customer satisfaction and retention.

3. Mental Health Monitoring : Emotion recognition can be used in mental health apps to monitor users' emotional states over time. This can help in early detection of issues like depression or anxiety, prompting timely interventions.

4. Education : In e   learning platforms, recognizing students' emotions can help tailor the learning experience. For example, if a student is confused or frustrated, the system can provide additional resources or adjust the difficulty level.

5. Entertainment : In gaming, emotion recognition can be used to adapt the game experience based on the player's emotional state, making games more engaging and personalized.

6. Smart Homes : Integrating emotion recognition in smart home systems can enhance user experience. For example, if the system detects that a user is stressed, it can adjust the lighting, play calming music, or suggest relaxation techniques.

7. Automotive Industry : In cars, emotion recognition can improve safety by detecting driver fatigue or stress and providing alerts or suggestions to take a break.

These applications show how speech emotion recognition can make technology more responsive and personalized, improving user experience across various do s.

Key Features:

Main Functionalities and Components of the Speech Emotion Recognition Project

1. **Data Collection and Preprocessing**:
   - **Data Collection**: Gather audio datasets containing speech samples labeled with corresponding emotions. Common datasets include the Ryerson Audio Visual Database of Emotional Speech and Song (RAVDESS) and the Interactive Emotional Dyadic Motion Capture (IEMOCAP) database.
   - **Preprocessing**: Clean and prepare the audio data by removing noise, normalizing volume levels, and segmenting the audio into manageable chunks. Extract relevant features such as Mel Frequency Cepstral Coefficients (MFCCs), chroma features, and spectral contrast using libraries like librosa.
2. **Exploratory Data Analysis (EDA)**:
   - **Visualization**: Use visualizations to understand the distribution of emotions in the dataset, the duration of audio samples, and the characteristics of extracted features. Libraries like matplotlib and seaborn can be helpful here.
   - **Statistical Analysis**: Perform statistical analysis to identify patterns and correlations between different features and emotions. This helps in understanding the data better and identifying any potential issues or biases.
3. **Model Development and Evaluation**:
   - **Model Selection**: Choose appropriate machine learning models for emotion recognition. Common choices include Convolutional Neural Networks (CNNs), Long Short Term Memory networks (LSTMs), and hybrid models combining both.
   - **Training**: Train the selected models using the preprocessed data. This involves splitting the data into training and validation sets, tuning hyperparameters, and using techniques like cross validation to ensure robustness.
   - **Evaluation**: Evaluate the model's performance using metrics such as accuracy, precision, recall, and F1 score. Visualize the results using confusion matrices and ROC curves to understand the model's strengths and weaknesses.
   - **Optimization**: Optimize the model by fine tuning hyperparameters, experimenting with different architectures, and using techniques like data augmentation to improve performance.

These components work together to create a robust speech emotion recognition system capable of accurately identifying emotions from audio signals. If you need more details on any specific component, feel free to ask!

Technology stock:

Programming language

- **Python :** Widely used for machine learning and analysis

1**. Pandas**

Purpose: Data manipulation and analysis.

Description: Pandas is a powerful Python library that provides data structures like DataFrames and Series, which are essential for handling and analyzing structured data. It offers a wide range of functions to manipulate, clean, filter, group, and merge datasets, making it a go to tool for data analysis tasks.

2.NumPy

Purpose:Numerical computations.

Description: NumPy stands for Numerical Python and is the foundation for many other scientific computing libraries in Python. It provides support for large, multi dimensional arrays and matrices, along with a vast collection of mathematical functions to operate on these arrays. NumPy is highly optimized and serves as the backbone for many data processing tasks.

3. Scikit learn

Purpose: Machine learning algorithms.

Description: Scikit learn is a popular library for implementing machine learning algorithms. It provides simple and efficient tools for data mining and data analysis, including classification, regression, clustering, and dimensionality reduction. It's widely used for building predictive models and for feature selection, model evaluation, and hyperparameter tuning.

4.TensorFlow/Keras

Purpose: Building and training deep learning models.

Description: TensorFlow is an open source deep learning framework developed by Google, and Keras is its high level API that simplifies building and training deep learning models. TensorFlow is used for creating machine learning models, especially deep neural networks, and it’s capable of running on CPUs, GPUs, and even TPUs, making it suitable for large scale machine learning tasks.

5. Librosa

Purpose: Audio processing.

Description: Librosa is a Python package specifically designed for analyzing and processing audio signals. It provides tools for tasks such as feature extraction, music analysis, and audio visualization. Librosa is widely used in music information retrieval (MIR) and audio analysis projects.

6. Matplotlib/Seaborn

PurposeData visualization.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is highly customizable and can be used to generate plots, histograms, bar charts, scatter plots, and more.

Seaborn is built on top of Matplotlib and provides a high level interface for drawing attractive and informative statistical graphics. It simplifies the creation of complex visualizations and is often used for exploring data relationships.

Development process:

Speech Emotion Recognition: Development Process

1. Planning

Problem Definition: The goal of the Speech Emotion Recognition (SER) project is to develop a system that can accurately identify and classify emotions from spoken audio. This requires understanding the nuances of vocal expressions and mapping them to specific emotional states, such as happiness, sadness, anger, or fear.

Research & Planning: Initial research focuses on existing SER systems, understanding the challenges, and reviewing relevant literature. The planning phase involves defining the project scope, selecting the appropriate methodologies, and setting clear, measurable goals for model performance.

2. Data Collection & Preprocessing

Data Sources: Speech data for SER is typically collected from publicly available datasets like the Ryerson Audio Visual Database of Emotional Speech and Song (RAVDESS) or custom datasets recorded in controlled environments. These datasets include audio clips labeled with corresponding emotions.

Cleaning: The audio data undergoes cleaning to remove noise, silence, and any irrelevant sections that might affect model performance. This process may include techniques like trimming, normalization, and filtering.

Preprocessing: Preprocessing involves converting raw audio into a format suitable for analysis, such as extracting features like Mel Frequency Cepstral Coefficients (MFCCs), chroma features, and spectral contrast. These features help in capturing the emotional characteristics of speech.

3. Exploratory Data Analysis (EDA)

Insights Gained: EDA involves analyzing the distribution of emotions in the dataset, understanding the variance in audio features across different emotions, and identifying potential biases in the data. Visualization techniques, such as heatmaps, box plots, and pair plots, are used to explore relationships between features and emotions.

Summary Statistics: Summary statistics provide insights into the frequency and duration of different emotions in the dataset. For instance, mean, median, and standard deviation of key features like pitch, intensity, and rhythm might be calculated to understand their influence on emotion recognition.

4. Model Development

Model Selection: Various models are tested, including classical machine learning models like Support Vector Machines (SVM) and Random Forest, as well as deep learning models like Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

Implementation: Models are implemented using frameworks like TensorFlow/Keras and Scikit   learn. The models are trained on the preprocessed features extracted from the audio data.

Hyperparameter Tuning: Hyperparameters such as learning rate, number of layers, and activation functions are tuned using techniques like grid search or random search to optimize model performance.

5. Model Evaluation

Evaluation Metrics: Metrics like accuracy, precision, recall, F1   score, and confusion matrix are used to evaluate the model's performance. Cross   validation techniques, such as k   fold cross   validation, are employed to ensure the model's generalizability and robustness.

Validation Techniques: The model is tested on a separate validation set to measure its ability to generalize to unseen data. Techniques like stratified sampling may be used to ensure that all emotions are equally represented in the training and validation sets.

6. Deployment

Deployment Process: The final model is deployed as a real   time application or a web service, possibly using platforms like Flask, Django, or cloud services like AWS or Google Cloud. The deployment phase also involves setting up an API for real   time emotion detection from speech input.

Maintenance: Continuous monitoring is implemented to track model performance in production, ensuring it remains accurate and reliable. Regular updates and retraining might be necessary as more data becomes available or as the system is exposed to new variations in speech patterns.

Challenges and solutions :

Speech Emotion Recognition: Challenges & Solutions

1. Data Challenges

   Challenge: Limited and Imbalanced Datasets

   Explanation: One of the primary challenges in Speech Emotion Recognition (SER) is the availability of labeled data. Many datasets are small, and the distribution of emotions can be highly imbalanced, with certain emotions like happiness or sadness being overrepresented, while others like fear or disgust are underrepresented.

   Solution: Data Augmentation and Synthetic Data

   Resolution: To address data scarcity and imbalance, data augmentation techniques like pitch shifting, time stretching, and adding noise were applied to create more training samples. Additionally, synthetic data generation using techniques like GANs (Generative Adversarial Networks) or oversampling methods like SMOTE (Synthetic Minority Over sampling Technique) helped balance the dataset.

2. Feature Extraction Challenges

   Challenge: Capturing Relevant Features

   Explanation: Emotion in speech is conveyed through various vocal features, but identifying and extracting the most relevant features that effectively capture emotional nuances can be difficult. Features like MFCCs, pitch, and energy may not fully represent the complex emotional state in speech.

   Solution: Hybrid Feature Sets and Deep Learning

   Resolution: A combination of traditional feature extraction methods (e.g., MFCCs, chroma) and deep learning based approaches was used. CNNs were employed to automatically learn feature representations directly from the raw audio waveforms, capturing intricate patterns that manual feature extraction might miss.

3. Model Performance Challenges

   Challenge: Overfitting and Generalization

   Explanation: Deep learning models, particularly when dealing with small datasets, are prone to overfitting, where the model performs well on the training data but fails to generalize to new, unseen data. This was especially problematic when training complex models like RNNs or CNNs.

   Solution: Regularization and Cross Validation

Resolution: Techniques such as dropout, L2 regularization, and batch normalization were applied to reduce overfitting. Additionally, k fold cross validation was used to assess model performance across different subsets of the data, ensuring that the model generalized well to new data.

4. Deployment Challenges

Challenge: Real Time Processing and Latency

Explanation: Deploying an SER model in a real time application presents challenges related to processing speed and latency. The system must quickly analyze speech input and return emotion predictions without significant delays, which can be challenging given the computational complexity of deep learning models.

Solution: Model Optimization and Scalable Infrastructure

Resolution: The model was optimized for speed using techniques like model pruning, quantization, and using lighter architectures. Additionally, deploying the model on scalable cloud infrastructure with GPU support ensured that the system could handle real time processing demands efficiently.

5. Interpretability Challenges

Challenge:Understanding Model Decisions

Explanation: Deep learning models, while powerful, often act as "black boxes," making it difficult to understand how they arrive at a particular emotion prediction. This lack of interpretability can be a barrier to trust and adoption in sensitive applications like customer service or healthcare.

Solution:Explainable AI (XAI) Techniques

Resolution:Techniques like LIME (Local Interpretable Model agnostic Explanations) or SHAP (SHapley Additive exPlanations) were used to provide insights into the model's decision making process. By analyzing the contribution of different features to the final prediction, these tools helped in making the model's behavior more transparent and understandable.

6. Cultural and Linguistic Variations

Challenge:Handling Diverse Speech Patterns

Explanation: Speech emotion recognition systems often struggle with variations in accents, dialects, and cultural expressions of emotion, which can significantly affect the accuracy of the model across different user populations.

Solution:Transfer Learning and Domain Adaptation

Resolution: Transfer learning techniques were employed, where the model was pre trained on a large, diverse dataset and then fine tuned on specific datasets representing different accents or languages. Domain adaptation strategies were also used to adjust the model to perform better across different linguistic and cultural contexts.

Learning:

Speech Emotion Recognition: Key Learnings

1. Understanding the Complexities of Audio Data

Learning:Deep Dive into Audio Signal Processing

Explanation: Working on Speech Emotion Recognition (SER) provided an in depth understanding of audio signal processing techniques. Key learnings included how to handle raw audio data, extract meaningful features like MFCCs, chroma features, and spectrograms, and the importance of preprocessing steps such as noise reduction and normalization. This project highlighted the intricate relationship between audio features and the emotional content of speech.

2. Mastering Feature Engineering and Selection

Learning: Importance of Feature Engineering in SER

Explanation:The project emphasized the critical role of feature engineering in SER. Identifying the right features that capture emotional nuances was essential for model performance. The experience reinforced the importance of both traditional feature extraction methods and leveraging deep learning to automatically learn features. It also underscored the significance of feature selection techniques to improve model accuracy and reduce overfitting.

3. Gaining Expertise in Model Development and Tuning

Learning: Model Selection and Hyperparameter Tuning

Explanation: The project provided hands on experience in developing and fine tuning machine learning models. Experimenting with different algorithms ranging from traditional models like SVMs to deep learning models like CNNs and RNNs offered insights into their strengths and limitations in the context of SER. Hyperparameter tuning, using methods like grid search and random search, highlighted the impact of fine tuning on model performance and generalization.

4. Enhancing Skills in Model Evaluation

Learning: Critical Evaluation Metrics

Explanation: Understanding and applying the right evaluation metrics was a key takeaway. Metrics like accuracy, precision, recall, and F1 score were crucial for assessing model performance. Additionally, the use of cross validation techniques to ensure the model's robustness and generalizability was a valuable lesson. This experience reinforced the importance of a rigorous evaluation process in developing reliable machine learning models.

Exploring Explainability in Machine Learning

Learning: Interpretability of Complex Models

Explanation: The project underscored the importance of model interpretability, particularly in applications like SER where understanding the model's decision making process is critical. Learning to use tools like LIME and SHAP to explain model predictions provided valuable experience in making complex models more transparent and trustworthy, which is increasingly important in AI driven applications.

6. Tackling Deployment Challenges

Learning:Model Optimization and Deployment

Explanation: Deploying the SER model in a real time environment required learning about model optimization techniques, such as pruning and quantization, to ensure low latency and high performance. Gaining experience in deploying models on cloud platforms and setting up scalable infrastructure for real time applications was a significant learning outcome. This project emphasized the importance of considering deployment constraints early in the development process.

7. Adapting to Cultural and Linguistic Variations

Learning: Handling Diversity in Speech

Explanation: The project highlighted the challenges associated with cultural and linguistic variations in speech. Learning to apply transfer learning and domain adaptation techniques to address these challenges was crucial. This experience provided insights into the need for diverse and representative datasets to build more inclusive and robust models that perform well across different populations.

8. Project Management and Collaboration

Learning: Effective Project Planning and Teamwork

Explanation: Beyond technical skills, the project reinforced the importance of careful planning, documentation, and collaboration in a multidisciplinary team. Managing the project from planning to deployment required strong organizational skills, effective communication, and the ability to work closely with others, including data engineers, domain experts, and software developers.

This project in Speech Emotion Recognition not only deepened technical expertise in data science and machine learning but also provided valuable experience in overcoming practical challenges and enhancing the applicability of AI models in real world scenarios.

Future scope:

Speech Emotion Recognition: Future Scope

1. Improving Model Accuracy and Generalization

Potential Improvement:Advanced Deep Learning Architectures

Description:Future work could explore more advanced deep learning architectures like Transformers, which have shown promise in natural language processing tasks. Transformers could be adapted for SER to better capture the temporal dependencies in speech and enhance the model’s ability to generalize across diverse speech patterns and emotions.

Additional Model:Ensemble Learning

Description: Implementing ensemble learning techniques, such as combining multiple models (e.g., CNNs, RNNs, and SVMs), could improve overall performance by leveraging the strengths of different algorithms. Ensemble methods can reduce model variance and improve robustness, leading to more accurate emotion recognition.

2. Handling Diverse and Real World Data

Potential Improvement: Larger and More Diverse Datasets

Description: Collecting and curating larger, more diverse datasets that include a wider range of languages, accents, and emotional expressions would enhance the model's ability to handle real world scenarios. This would involve gathering data from various cultural and linguistic backgrounds to ensure the model is inclusive and performs well across different populations.

Further Analysis: Transfer Learning and Domain Adaptation

Description: Applying transfer learning to adapt models pre trained on large datasets like Common Voice to specific SER tasks could significantly improve performance. Domain adaptation techniques could also be used to fine tune the model for different environments or specific user groups, such as adapting the model for use in different regions or industries.

3. Enhancing Model Interpretability

Potential Improvement:Incorporation of Explainable AI Techniques

Description: As SER models become more complex, enhancing interpretability is crucial. Future work could focus on integrating advanced Explainable AI (XAI) techniques that provide more granular insights into how the model arrives at specific emotion predictions. This could involve developing new visualization tools that make the decision making process more transparent to end users.

Further Analysis:Emotion Pathways

Description: Analyzing the pathways through which different features contribute to emotion recognition could provide deeper insights into the model's workings. Understanding how various audio features, like pitch, rhythm, and tone, interact to signal different emotions could lead to more targeted improvements in model architecture and feature selection.

4. Real Time Performance Optimization

Potential Improvement:Edge Computing and On Device Processing

Description: To reduce latency and improve the real time performance of SER systems, future work could explore edge computing solutions, where the model is deployed directly on devices like smartphones or embedded systems. This would reduce the dependency on cloud infrastructure and enable faster, more responsive emotion recognition.

Additional Model:Lightweight and Efficient Models

Description: Developing lightweight models that require less computational power without compromising accuracy could make SER systems more accessible and scalable. Techniques like model pruning, quantization, and knowledge distillation could be further explored to create efficient models suitable for deployment in resource constrained environments.

Multimodal Emotion Recognition

Potential Improvement:Integration of Multimodal Data

Description: Future work could involve combining speech data with other modalities, such as facial expressions, body language, and physiological signals, to create a more comprehensive emotion recognition system. Multimodal emotion recognition has the potential to significantly enhance accuracy by leveraging different sources of emotional cues.

Further Analysis:Cross Modal Learning

Description: Exploring cross modal learning techniques, where the model learns to correlate and integrate features from different modalities, could provide richer and more context aware emotion recognition. This could involve the development of architectures that can process and fuse audio, video, and other sensor data in real time.

6. Personalization and Context Awareness

Potential Improvement:Context Aware Emotion Recognition

Description: Incorporating contextual information, such as the speaker’s environment, background noise, or even their recent interaction history, could improve the accuracy of emotion recognition. Context aware models could better interpret emotions by considering the situational context in which the speech occurs.

Further Analysis: Personalized Emotion Recognition

Description: Future developments could focus on personalizing SER systems to individual users by adapting the model to their unique speech patterns and emotional expressions. This could involve continuous learning algorithms that update the model as it interacts with a specific user, leading to more accurate and user specific emotion detection.

7. Ethical Considerations and Bias Mitigation

Potential Improvement: Bias Detection and Mitigation

Description: Addressing ethical concerns related to bias in SER systems is a critical area for future research. Developing techniques to detect, measure, and mitigate bias in emotion recognition models, especially across different demographic groups, would enhance the fairness and reliability of these systems.

Further Analysis: Ethical AI Frameworks

Description:Incorporating ethical AI frameworks into the development process could ensure that SER systems are designed with fairness, transparency, and accountability in mind. This might involve regular audits of the model's performance across different user groups and implementing feedback mechanisms to continually improve its ethical standards.

These potential improvements and extensions highlight the ongoing opportunities for enhancing Speech Emotion Recognition systems, making them more accurate, robust, and applicable to a wide range of real world scenarios.

Conclusion:

The Speech Emotion Recognition (SER) project represents a significant achievement in the field of data science, combining advanced techniques in audio processing, machine learning, and deep learning to create a system capable of identifying emotions from speech with high accuracy. This project not only demonstrated the feasibility of using modern data science tools to tackle complex problems in human computer interaction but also highlighted the potential for SER systems to transform various industries, including customer service, healthcare, and entertainment

Code:

```
import pandas as pd
```

```
import numpy as np

import os

import seaborn as sns

import matplotlib.pyplot as plt

import librosa

import librosa.display

from IPython.display import Audio

import warnings

warnings.filterwarnings('ignore')


# Load the Dataset from the specified path

dataset_path = r"C:\Users\Asus\OneDrive\Desktop\TESS Toronto emotional speech set data"

paths = []

labels = []

for dirname, _, filenames in os.walk(dataset_path):

    for filename in filenames:

        paths.append(os.path.join(dirname, filename))

        label = filename.split('_')[   1]

        label = label.split('.')[0]

        labels.append(label.lower())

    if len(paths) == 2800:

        break

print('Dataset is Loaded')
```

```
# Check dataset length
len(paths)

# Display first 5 paths
print(paths[:5])

# Display first 5 labels
print(labels[:5])

# Create a dataframe
df = pd.DataFrame()
df['speech'] = paths
df['label'] = labels
df.head()

# Value counts of labels
df['label'].value_counts()

# Exploratory Data Analysis
sns.countplot(data=df, x='label')

# Functions to plot waveplot and spectrogram
```

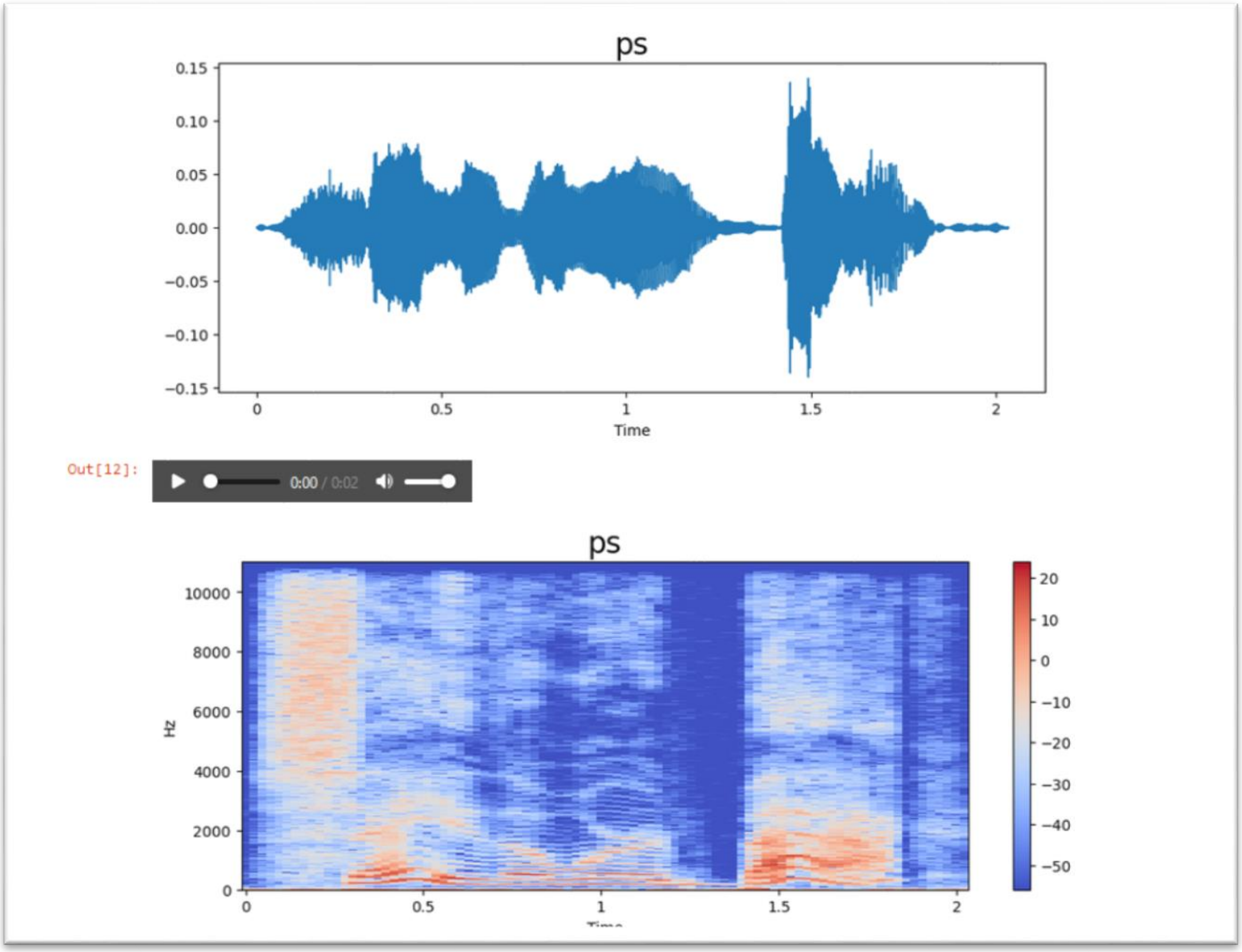
```
def waveplot(data, sr, emotion):

    plt.figure(figsize=(10,4))

    plt.title(emotion, size=20)

    librosa.display.waveshow(data, sr=sr)

    plt.show()


def spectogram(data, sr, emotion):

    x = librosa.stft(data)

    xdb = librosa.amplitude_to_db(abs(x))

    plt.figure(figsize=(11,4))

    plt.title(emotion, size=20)

    librosa.display.specshow(xdb, sr=sr, x_axis='time', y_axis='hz')

    plt.colorbar()


# Example usage of waveplot and spectrogram functions

emotion = 'fear'

path = np.array(df['speech'][df['label']==emotion])[0]

data, sampling_rate = librosa.load(path)

waveplot(data, sampling_rate, emotion)

spectogram(data, sampling_rate, emotion)

Audio(path)


# Feature Extraction
```

```
def extract_mfcc(filename):
    y, sr = librosa.load(filename, duration=3, offset=0.5)
    mfcc = np.mean(librosa.feature.mfcc(y=y, sr=sr, n_mfcc=40).T, axis=0)
    return mfcc

# Apply MFCC extraction to the dataset
X_mfcc = df['speech'].apply(lambda x: extract_mfcc(x))

# Convert to numpy array
X = [x for x in X_mfcc]
X = np.array(X)
X.shape

# Input split
X = np.expand_dims(X,    1)
X.shape

# One   hot encoding of labels
from sklearn.preprocessing import OneHotEncoder
enc = OneHotEncoder()
y = enc.fit_transform(df[['label']])
y = y.toarray()
y.shape
```

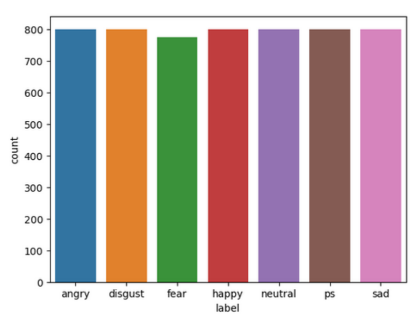
```
# Create the LSTM Model
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout

model = Sequential([
    LSTM(256, return_sequences=False, input_shape=(40,1)),
    Dropout(0.2),
    Dense(128, activation='relu'),
    Dropout(0.2),
    Dense(64, activation='relu'),
    Dropout(0.2),
    Dense(7, activation='softmax')
])

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.summary()

# Train the model
history = model.fit(X, y, validation_split=0.2, epochs=50, batch_size=64)

# Plot the results
epochs = list(range(50))
```

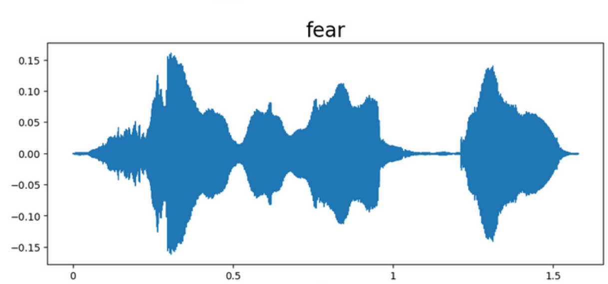
```
acc = history.history['accuracy']

val_acc = history.history['val_accuracy']


plt.plot(epochs, acc, label='train accuracy')

plt.plot(epochs, val_acc, label='val accuracy')

plt.xlabel('epochs')

plt.ylabel('accuracy')

plt.legend()

plt.show()


loss = history.history['loss']

val_loss = history.history['val_loss']


plt.plot(epochs, loss, label='train loss')

plt.plot(epochs, val_loss, label='val loss')

plt.xlabel('epochs')

plt.ylabel('loss')

plt.legend()

plt.show()
```

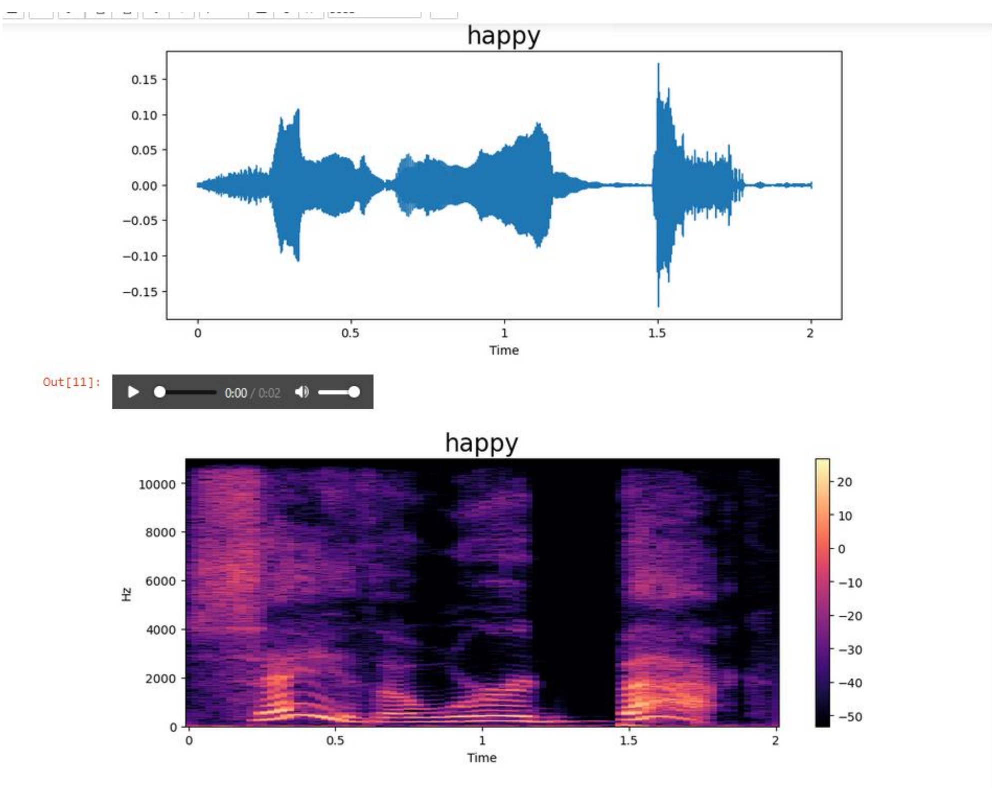
SCREENSHOTS:

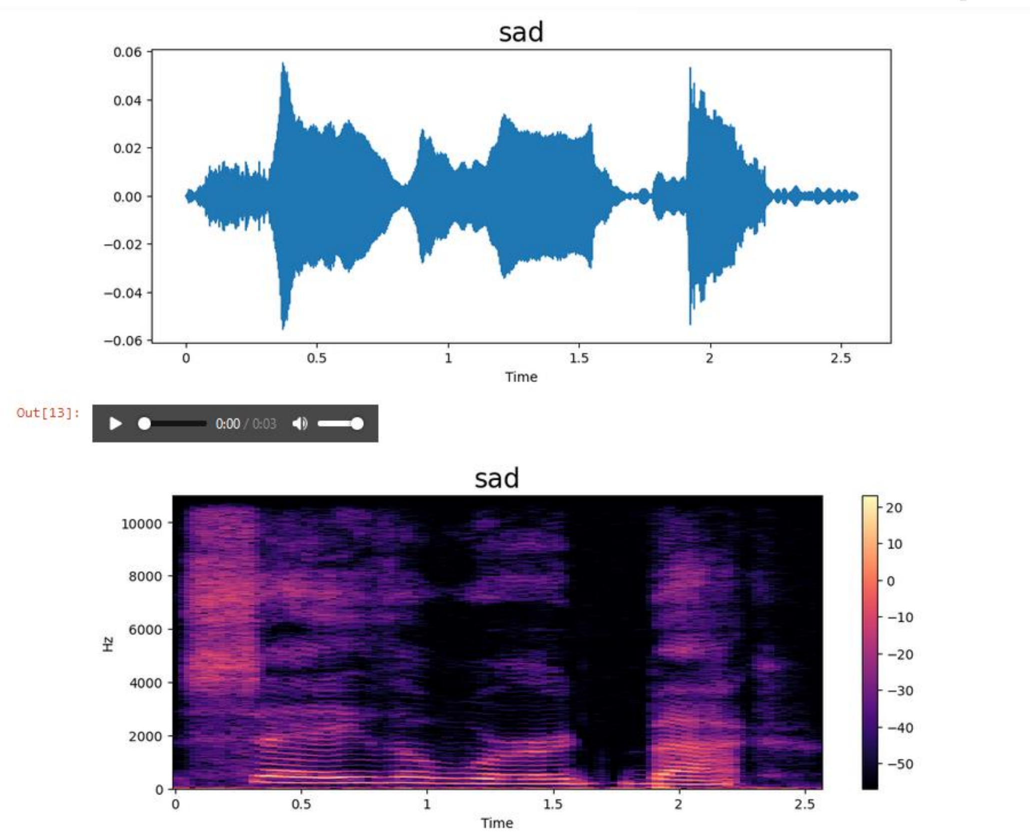

ps
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
0
0.5
1
1.5
2
Time
Out[12]:
0:00 / 0:02
ps
Hz
10000
8000
6000
4000
2000
0
20
10
0
−10
−20
−30
−40
−50

+


800
700
600
500
400
300
200
100
0
count
angry
disgust
fear
happy
neutral
ps
sad
label


fear
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
0
0.5
1
1.5





DEMO/PRESENTATION:

https://www.linkedin.com/posts/shiramdasu-abhiram-4b3924304_dataanalytics-speechrecognition-machinelearning-activity-7232376526355283968-V2Ba?utm_source=share&utm_medium=member_android