

Emotion Analysis in Audio conversation

*A Major Project Report Submitted in Fulfilment
of the Requirements of the course*

**Machine Learning
MECE 6397 (29929)**

for

Semester 1 (Fall 2021)

by

SUDHARSAN RAGOTHAMAN-2097435

Under the Guidance of

Prof. Gangbing Song



**University of Houston
Batch: 2021-2023**

● ACKNOWLEDGEMENT

First and foremost, we are truly indebted to our supervisor Prof. Gangbing Song for his invaluable supervision throughout the course of our study, without which this project would not be in the present form.

I also thank him for his excellent guidance and insightful feedback, which helped us to bring our research work to a higher level.

I extend heartfelt gratitude to everyone in the panel, for the successful completion of this project.

Sudharsan Ragothaman

● ABSTRACT

Recent improvements in the online market of products increased the need of customer care both quality wise and quantity wise, deep learning-based audio processing models have achieved a huge breakthrough in not only processing the clarity in an audio but also in the area of finding emotions of customers and training a machine to respond to customer queries according to their response and sentiment towards the call. The current methodologies use advanced machine learning models to do audio readings but still, there is a gap in producing realistic consistent outputs. My work proposes a novel method of predicting Sentiment of speaker, using multiple models like KNN,SVM and random forest by diarizing the audio clips into different chunks and clustering the audio. The trained model was used to classify the sentiment of every chunk of the audio clip. Also, audio is converted to text to predict the emotion in words used. In this way accuracy of the project is multi fold as multiple aspects of emotions are considered.

● CONTENTS

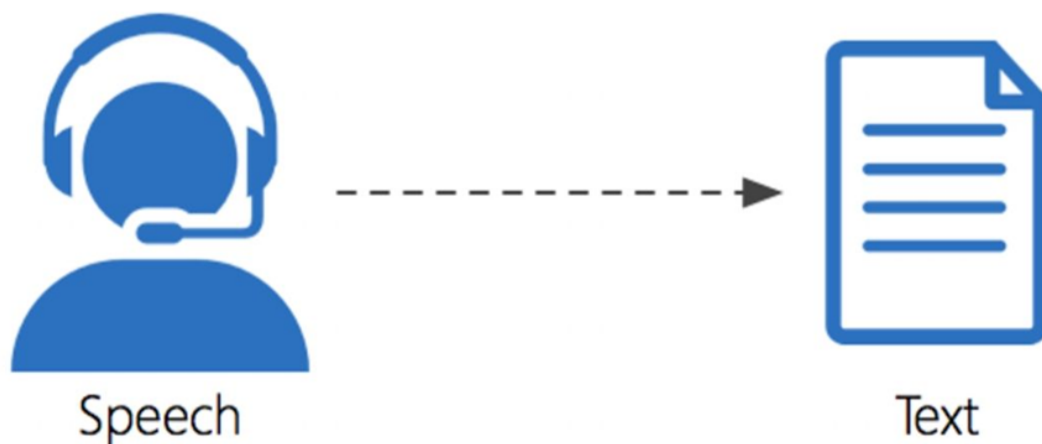
ACKNOWLEDGEMENT	i
ABSTRACT	ii
CONTENTS	iii
Chapter 1: INTRODUCTION	1
1.1 Introduction	1
1.2 Literature Review	2
1.3 Objective	5
1.4 Dataset	5
Chapter 2: MACHINE LEARNING	6
2.1 Overview of Machine Learning	6
2.2 Logistic Regression	6
2.3 Support Vector Machines	7
2.4 Decision Tree	7
2.5 Random Forest Algorithm	8
Chapter 3: EXPERIMENTATION METHODOLOGY	9
3.1 Overview	9
3.2 Speaker Diarization	10
3.3 Audio to Text Conversion	11
3.4 Classification of Audio Samples	12
3.5 Classification of Text Samples	13
Chapter 4: SIMULATION AND RESULTS	14
4.1 Result of Speaker Diarization	14
4.2 Result Audio to Text Conversion	14
4.3 Simulation of audio samples	15

4.4 Simulation of Text samples	17
Chapter 5: FUTURE WORKS	19
REFERENCES	20

- **Chapter 1: INTRODUCTION**

1.1 Introduction

The goal of this research work is to use machine learning algorithms to predict the emotion of a person who is contacting customer care for their product queries. In this project, the audio is converted into text using Google API. This text sentiment is trained using multiple supervised learning models like SVM, linear SVM and Decision tree and the accuracies are compared in which Decision Tree has the best accuracy. With this the converted text from the audio is tested.



Also, Voice Activation Detection is used to segment the audio into different chunks and classify the customers emotions separately by removing the caller's audio out. Once the audio is segmented, speaker identification is done using MAP estimation on every chunk using Universal Background Model (UBM) and Gaussian Mixture Model (GMM). Every chunk represents a different GMM while UBM represents a GMM on

the whole audio file. Speech clustering is done using spectral clustering on every audio segment.



In this way the accuracy of the project is multifolded as multiple aspects of emotion is considered and the hybrid classifier is used to solve the problem statement.

1.2 Literature Review

Lately, there has been growing exploration add the sector of emotion recognition supported speech data to extend the delicacy of similar systems [1]. Still, only many of those attempts indeed increase the effectiveness of literacy models for speech data. Authors of [1] have correctly conceded that, although multitudinous bracket styles [2-5] are tested and habit to facilitate the effectiveness of the training model, HMM remains the most common and effective system. The rigor of HMM on colorful datasets compared to GMM, ANN, and SVM, etc., is still similar. Also, HMM presents the advantage and skill to reuse successional data, e.g., processing of frame-position features, whereas GMM, ANN, and SVM warrant this and can not reuse sequences of point vectors.

HMM, still presents some limitations as conceded in [6-8]. The main limitations of HMM is its propagative nature and thus the neutrality thesis among its countries and the interpretations. A new model called the maximum entropy Markov model was proposed to rectify these limitations. This model shows better results for specific operations/ tasks, including part-of-speech trailing (POS) [9], substantiation abstraction [10], and the recognition of automatic speeches [11].

Some attributes may degrade the delicacy, e.g., the birth of the prominent features and high similarity among different feelings in the presence of low between-class friction in the point space. [12] have presented a new interpretation of the HCRF algorithm that uses full covariance Gaussian viscosity functions. Also, it proved theoretically and experimentally that the recognition rates of the proposed approach are comparatively precise than being algorithms.

Current exploration has stressed new approaches to emotion bracket from speech-supported audio features. The work by Shen. et. al [21] fete five different emotion countries like nausea, lethargy, sadness, neutral, and happiness using the features pitch, energy, LPCC, MFCC, and LPCMCC. They have explained and compared different combinations of features. Berlin emotional database added an advantage to their work. They got additional delicacy in various combinations of features. In their work [16] Casale. et. al; described working in a DSR terrain. Characteristics considered for this trial were uprooted by using ETSI ES 202 211V.1.1.1 S standard front. In preparation, they used two different speech corpora EMO-DB in German and SUSAS in American English. Their result showed that Support Vector

Mishne [13] classified the blog textbook harmoniously with the mood reported by its author during the memo. Mishne considered different textual features like frequency counts of words, the emotional opposition of posts, length of posts, PMI, emphasized words and special symbols like emoticons and punctuation marks. PMI-Point-wise Mutual Information provides a numerical weight for keywords related to a particular mood. SVM (Support Vector Machine) classifier was chosen for the bracket. Text mining over transcribed audio recordings was performed in [14] to find the speaker's emotion. The dataset (audio discussion of the guests) for this trial was collected from a call center. The experimenters used different point selection styles in this work. The unsupervised and supervised system further clarified the textbook bracket. The substantiation demonstrated

The mongrel approach, which mixes textbook mining and speech mining, is extensively employed in the sector of musical kidney bracket [15], [16], [17]. In the

case of the music mood bracket, lyrics and audio features were used to ameliorate the delicacy of the bracket.

The work of [18] and [19] captures druggies' intention in the natural- language textbook query format using a pattern-matching approach. The use of the pattern-matching system is easy-to-use and straightforward. Nonetheless, words can have different meanings, causing the incorrect result of emotional reflections. In their paper, Yasmina et al. [20] identify feelings from YouTube commentary using an unsupervised machine learning algorithm, Support Vector Machine. It utilizes commentary from colorful videotape orders uprooted by using YouTube API. The algorithm works on word- position, which is also combined into an emotion bracket at the judgment position. It results in 92.75 as average perfection and 68.82 as an average delicacy. Likewise, Banik and Rahman [21] proposed feelings models from movie reviews with Naive Bayes and Support Vector Machine with N-Gram because of the features. As the pre-processing phase, stemming was also applied to the textbook to prize the base word. The stylish results achieved during this exploration were the F1 score of 86.00 and 83.00 for Support Vector Machine and Naive Bayes, independently. Hasan et al. [22] proposed Naive Bayes, Support Vector Machine, and Decision Tree to fete feelings from reused labeled textbook from Twitter. The features used in this exploration were Unigram of word, emoticons, punctuation, and negation. The stylish results of the F1 score achieved during this exploration were 90.00 for Support Vector Machine (only unigram point), 90.00 for Decision Tree (using all features), and 90.00 for Naive Bayes (with all components).

Liu et al. [23] also proposed Extreme Learning Decision Tree, Support Vector Machine, and Back-propagation Neural Network for feelings recognition. The feelings were then classified into six introductory feelings. The results are 89.60, 87.20, 82.30 for Extreme Learning Decision Tree, Support Vector Machine, and Back-propagation Neural Network, independently, as the average bracket delicacy.

1.3 Objective

Usually in emotion classification, researchers consider the acoustic features alone. For strong emotions like anger and surprise, the acoustic features pitch and energy are both high. In such cases, it is very difficult to predict the emotions correctly using acoustic features alone. But, if we classify speech solely on its textual component, we will not obtain a clear picture of the emotional content. In the proposed hybrid approach we consider both text and audio features. Fig shows the framework for hybrid approach.

1.4 Dataset

- I am using the dataset from RAVDESS for classification purpose of audio chunks obtained. It consists of 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent. This portion of the RAVDESS contains 1440 files: 60 trials per actor x 24 actors = 1440. Speech emotions includes calm, happy, sad, angry, fearful, surprise, and disgust expressions. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression.
- Dataset consists of different emotion like - *neutral, calm, happy, sad, angry, fearful, disgust, surprised*
- the labeled descriptions for each dataset in the same order as the label categories in the datasets. For instance, if an observation in SCv1 has label 1, then that observation is in the sarcasm category. Multiple emotions like happy calm surprised were classified as positive and angry fearful disgust were classified as negative emotion.
- Twitter dataset of sentiment analysis of text documents is chosen. Assigning a sentiment score to each phrase and component (-1 to +1)

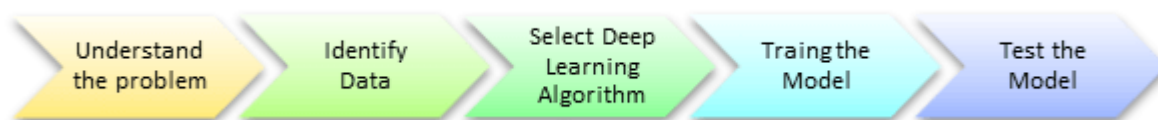
- **Chapter 2: MACHINE LEARNING**

2.1 Overview of Machine Learning

Machine learning is the science and art of programming computers so they can learn from the data. In other words, machine learning is the application of artificial intelligence and it gives the devices the ability to learn from their experiences, improve themselves and make the best possible decisions.

Ex: Shopping from any e-commerce or website suggests the related search like People who bought also saw this.

Deep Learning structures the algorithms into multiple layers in order to create an “artificial neural network”. This neural network can learn from the data and make intelligent decisions on its own. The working of deep learning process is depicted in the below image as shown.



2.2 Logistic Regression

Logistic Regression is a method in statistics that predicts value based on previous observations from a dataset. It is an essential tool in machine learning as it allows classifying a dataset based on history. The more the number of elements in the dataset, the algorithm gets more accurate. Logistic Regression can be used to classify a set of e-mails as spam or not spam, Online transactions as fraudulent or non-fraudulent based on a previous sample data set's learning. Logistic Regression can be binomial, consisting of two possible outcomes or multinomial. In binomial Logistic Regression, the outcomes are "1" or "0", "pass" vs. "fail" or "win" vs. "loss". Multinomial Regression comes in situations when output can have three or more possibilities.

For binary logistic Regression, the conditional probability of Y (dependent variable, given X (the independent variable), can be written as Probability of Y=1 given X or Probability of Y=0 given X.

The output values, $h_{\beta}(x_i)$, can be > 0 or < 1 , $0 \leq h_{\beta}(x_i) \leq 1$.

The hypothesis value is defined as,

$$h_{\beta}(x_i) = 1/g(\beta^T x)$$

where g is the sigmoid function.

2.3. Support Vector Machines

Support Vector Machine is a common supervised learning algorithm. It is used for classification and regression machine learning problems. The aim is to create a decision boundary that will separate the n -dimensional space from the database into separate classes. After the boundaries are created, any new data point can be categorised easily in the future.

The best decision boundary is the hyperplane, which is created by choosing the extreme points in the dataset. These points are known as support vectors.

Working of SVM:

Suppose we have two classes in a dataset, blue and green. It has two features x_1 and x_2 . These two classes can be classified by either the red line or the green line. There can be even more lines separating these two points. SVM in this case helps us to find the best decision boundary or the hyperplane. It finds the nearest points to the line from both the classes, which are the support vectors.

The margin or the distance between hyperplane and support vector is maximised using SVM, to give optimal hyperplane.

2.4. Decision Tree

A decision tree is a commonly used tool in strategy identification in data mining and machine learning. It uses a map of decisions resembling a tree and is a decision support tool which is used to analyse their possible consequences including resource costs, chance event outcomes, and utility.

Decision trees predict the outcomes of a process using a series of dependent questions. With the historical data of the process, the decision tree learns the best set of questions to ask along with the sequence of said questions. The questions are asked in decreasing order of importance, making a decision tree where the first question is at the root and terminals or leaves are outcomes and in between are split/decision nodes. Construction of a decision tree involves selection of attributes and conditions that will produce the tree. Irrelevant branches are then pruned which could reduce accuracy.

This involves identifying outliers which could give disproportionate weight to rare occurrences in the data.

2.5. Random Forest Algorithm

Random Forest is supervised machine learning technique, which is used for regression and classification problems in Machine Learning. It combines numerous classifiers to find the solution to complex machine learning problems and thereby improve the performance. It is based on ensemble learning.

It contains multiple decision trees on subsets of the dataset and takes the average to improve the accuracy. Instead of one decision tree, it takes prediction from each of them and based on the votes, it predicts final output.

- **Chapter 3: EXPERIMENTATION METHODOLOGY**

3.1 OVERVIEW

Our methodology is divided in 4 parts-

1. Source Separation of audio into different chunks [Separating customer audio]
2. Conversion of audio to text
3. Classification based on audio
4. Classification based on Text

The first phase of our system is segmentation of the audio into parts using Google's webrtc Voice Activation Detection . The splitted parts of audio are run through Adaptive MAP estimation to find the customer and remove the audio chunk of company caller using the generated super vectors.

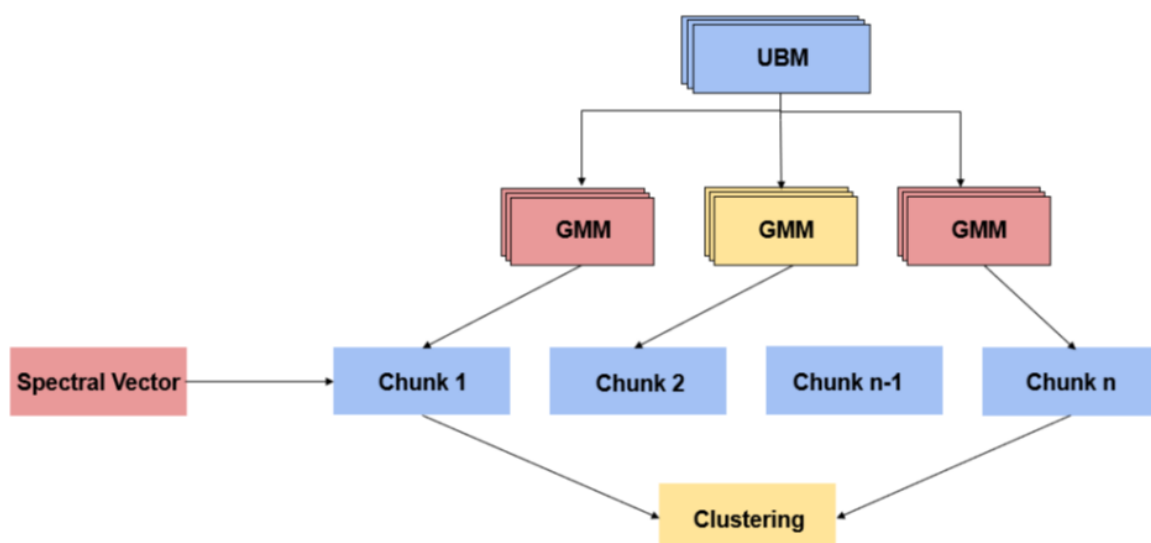
Once the audio file of conversation between the customer and call center agent is divided into chunks, we pass it through Google API to convert the customer chunk audio to text which divides our project into two arena.

Now the model is ready for classification, Using the RAVDESS emotion dataset the predictive sentiment analysis is run in to multiple classifier models to find the best accuracy of all. And the text is passed into text sentiment analysis model built using logistic regression trained with nplk twitter dataset to predict the emotion of the customer



3.2 SPEAKER DIARIZATION

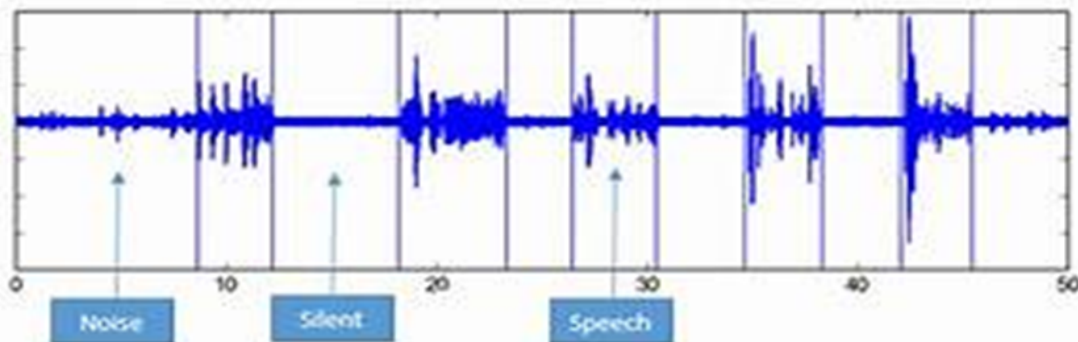
Google's webrtc Voice Activation Detection is used to segment the audio. Voice activation Detection involves in separating the voiced part from the unvoiced part. Passing some part of window over the whole audio with some addition overlap, by calculating the pause in the audio chunks are parted with window size of 32 ms and hop size of 10 ms.



The splitted parts of audio are run through Adaptive MAP estimation to find the customer and remove the audio chunk of company caller using the generated super vectors. Acoustic features such as MFCC, the adaptive MAP estimation for speaker identification are identified using the first order derivative along with second order derivative of the acoustic features.

We need to find the inconsistencies in the channel and better represent the variations between the two speakers. By the factor analysis approach, we convert the features into a super vector, a vector consisting of means trained using GMM. Since there are similar speech utterances in the whole audio, we train a Universal Background Model (UBM), a GMM on the whole audio file. The UBM is an independent speaker model. It represents features that are not dependent on the speaker. These feature vectors comprising mean and covariance are used to train the individual models. The means and prior probabilities from the UBM are used as initial values for each chunk's means and priors of the individual GMMs. The means of UBM are trained using the EM algorithm, in the M step, the means are updated using Adaptive Co-efficient.

The super vector obtained from the Speaker Identification is passed to different Clustering algorithms with affinity as cosine similarity between each chunk to cluster the Gaussian mixtures into two speakers.



3.3 Audio to Text conversion

Google API is used for audio to text transcription. The SoundFile library is used to read the audio file. It's a NumPy-based audio library that uses libsndfile, CFFI, and NumPy. The audio file is partitioned and sampled into 10-second audio snippets. To process the clips, a recogniser instance is established, and each clip is recorded using

the recogniser instance, with the audio data provided in a Google API readable format. The audio data is then fed into the main recognize google API. It uses the Google Speech Recognition API to perform speech recognition on audio data. Each audio clip's text is stitched together, and the entire audio text is transcribed.

WaveNet is a machine-learning-based algorithm that generates these text-to-speech audios. It's a deep neural network capable of making computers produce stunning human-like sounds. Google Speech to Text is backed by the WaveNet model implemented using TensorFlow.



3.4 Classification of audio samples

RAVDESS speech data set is used to train the audio lines. Features such as MFCC, STFT, Differ, Mel Spectrum, Chroma and Tonnetz are uprooted from the audio clips of the dataset. We mound all the features to get a point vector of 193 dimensions.

The extracted features are compiled into a pickle file feature and label to serialize all the truth values and is passed through models like K Nearest neighbors, Random Forest and Support Vector machine to find the best fitted model amongst the supervised learnings.

The trained model from these supervised learnings in the Training phase is used to test the sentiment of the chunks generated after source separation. Every chunk will yield sentiment which can help us understand the sentiment of the customer throughout the conversation

3.5 Classification of text samples

- These chunks after run through the audio-text classifier is taken as a text to be tested with trained sample of twitter dataset by breaking thr each text document down into its component parts (sentences, phrases, tokens and parts of speech)
- Identifying each sentiment-bearing phrase and component or a word
- Assign a sentiment score to each phrase and component (-1 to +1) considering the word used in the text
- Combining scores for multi-layered sentiment analysis

Implementing logistic regression for sentiment analysis on tweets. Deciding if the text is a positive comment or negative comment based on the sentiment score given.

Extracted features for logistic regression given some text. Implemented logistic regression from scratch to the text and applied logistic regression on a natural language processing task. Tested using your logistic regression with the accuracy on 99.5%

- **Chapter 4: SIMULATION AND RESULTS**

4.1 Speaker Diarization

Using an audio file with a telephonic conversation, audio files that have sentiment labelled for the whole conversation. The files have four emotion labels: sad, angry, happy, and neutral. Through VAD detection, segments of an audio file are separated. These segments contain voiced data, which are then passed on to the Adaptive MAP estimation process to obtain the super vector. After the super vector has been passed to the clustering, it will give the chunk numbers that belong to Speaker 0 and Speaker 1. We assign labels 0 and 1 based on which person starts to speak first, the first person will be labelled 0 and the other person will be labelled 1.

Sample audios after extraction

<https://drive.google.com/drive/folders/1LzXJvcGKnD3PwtaKuXyBnN-84Np5H5tw?usp=sharing>

4.2 Audio to text

The audio file is partitioned and sampled into 10-second audio snippets. To process the clips, a recogniser instance is established, and each clip is recorded using the recogniser instance, with the audio data provided in a Google API readable format. The audio data is then fed into the main recognize google API. It uses the Google Speech Recognition API to perform speech recognition on audio data. Each audio clip's text is stitched together, and the entire audio text is transcribed.

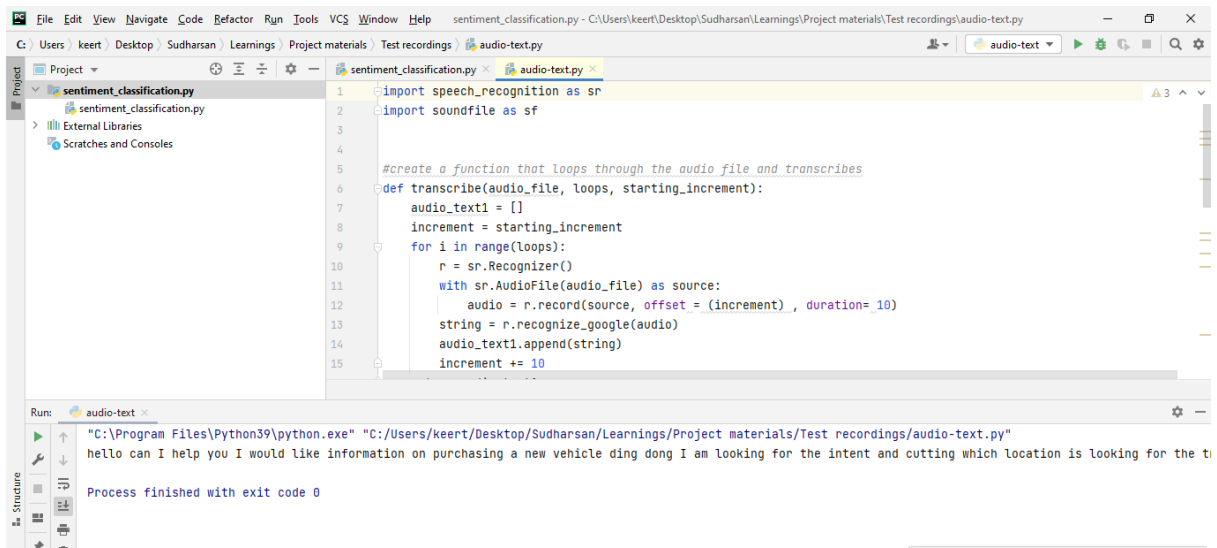
Sample Output

chunk-01.wav I am falling from market I am falling from market

chunk-06.wav Hanuman Mahima Mama ji great

chunk-09.wav Anjali Sharma spoken to nahin number in about drawing about school number about

chunk-14.wav Vikram 951 number phone number 951 number phone number



The screenshot shows an IDE window with a Python script named `audio-text.py`. The script imports `speech_recognition` as `sr` and `soundfile` as `sf`. It defines a function `transcribe` that takes `audio_file`, `loops`, and `starting_increment` as arguments. Inside the function, it initializes `audio_text1` as an empty list, sets `increment` to `starting_increment`, and enters a `for` loop that iterates `loops` times. In each iteration, it creates a `sr.Recognizer` object `r`, opens the `audio_file` as a source, records audio for a duration of 10 seconds starting from the `offset` (which is the `increment`), and then uses `r.recognize_google` to transcribe the audio. The resulting string is appended to `audio_text1`, and the `increment` is increased by 10. The script is shown running in the bottom console, which displays the output: "hello can I help you I would like information on purchasing a new vehicle ding dong I am looking for the intent and cutting which location is looking for the t". The process finished with exit code 0.

```
1 import speech_recognition as sr
2 import soundfile as sf
3
4
5 #create a function that loops through the audio file and transcribes
6 def transcribe(audio_file, loops, starting_increment):
7     audio_text1 = []
8     increment = starting_increment
9     for i in range(loops):
10         r = sr.Recognizer()
11         with sr.AudioFile(audio_file) as source:
12             audio = r.record(source, offset = (increment), duration=10)
13             string = r.recognize_google(audio)
14             audio_text1.append(string)
15             increment += 10
```

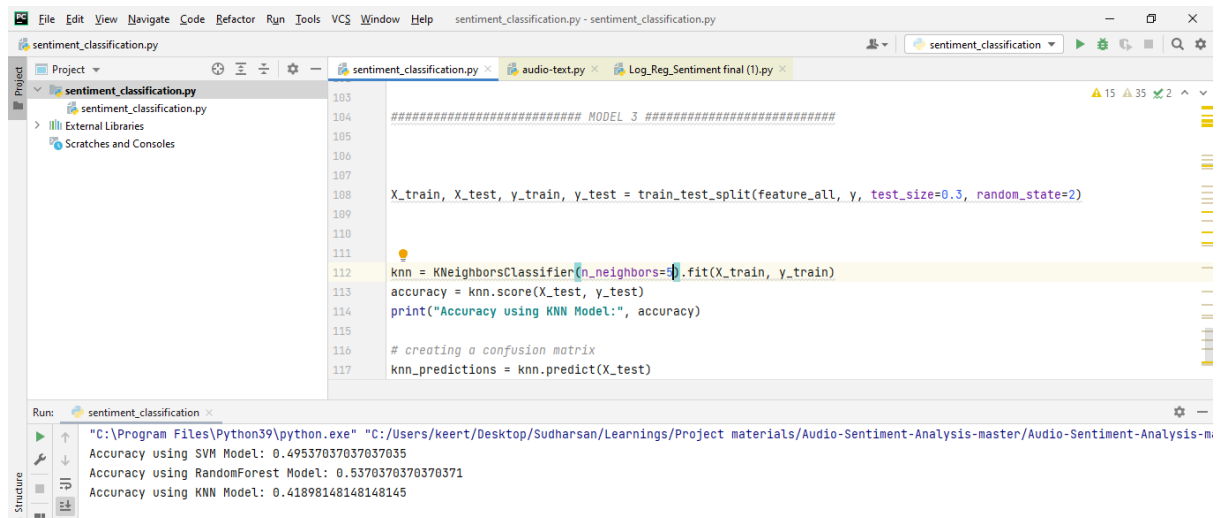
Run: audio-text
"C:\Program Files\Python39\python.exe" "C:/Users/keert/Desktop/Sudharsan/Learnings/Project materials/Test recordings/audio-text.py"
hello can I help you I would like information on purchasing a new vehicle ding dong I am looking for the intent and cutting which location is looking for the t
Process finished with exit code 0

4.3 Classification based on audio

Classification model was trained on the RAVDESS speech dataset. This dataset comprises of audio files from 24 actors. These actor utter a sentence in 8 emotions, happy , angry, calm, sad, surprised, neutral, disgust, fearful. We train our model by extracting acoustic features of the audio clip. We use a total of 193 features of all the audio clips from our RAVDESS dataset for classification purpose. We use two different algorithms for classifying the sentiment of our input chunk.

OUTPUT

1. SVM- kernel= “poly”; degree=10; c=1000
2. Random forest n_estimators=100, criterion='entropy', random_state=58
3. KNN, nearest neighbour=5



```

##### MODEL 3 #####

X_train, X_test, y_train, y_test = train_test_split(feature_all, y, test_size=0.3, random_state=2)

knn = KNeighborsClassifier(n_neighbors=5).fit(X_train, y_train)
accuracy = knn.score(X_test, y_test)
print("Accuracy using KNN Model:", accuracy)

# creating a confusion matrix
knn_predictions = knn.predict(X_test)

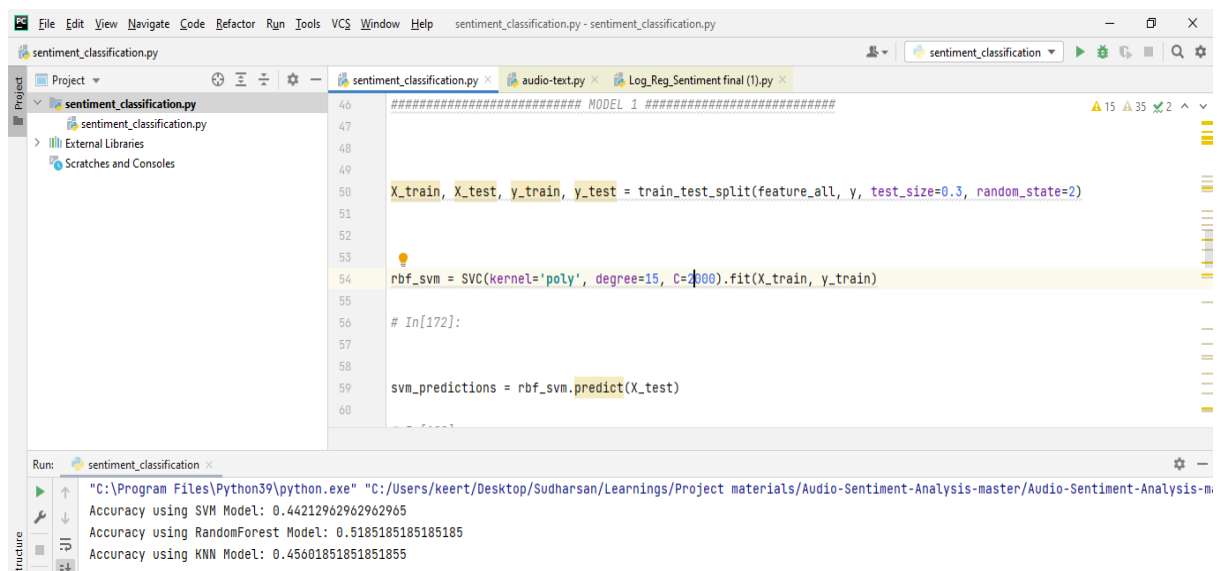
```

Run: sentiment_classification

"C:\Program Files\Python39\python.exe" "C:/Users/keert/Desktop/Sudharsan/Learnings/Project materials/Audio-Sentiment-Analysis-master/Audio-Sentiment-Analysis-m

Accuracy using SVM Model: 0.49537037037037035
 Accuracy using RandomForest Model: 0.5370370370370371
 Accuracy using KNN Model: 0.41898148148148145

4. SVM- kernel= “poly”; degree=15; c=2000
5. Random forest n_estimators=50, criterion='entropy', random_state=46
6. KNN, nearest neighbour=3



```

##### MODEL 1 #####

X_train, X_test, y_train, y_test = train_test_split(feature_all, y, test_size=0.3, random_state=2)

rbf_svm = SVC(kernel='poly', degree=15, C=2000).fit(X_train, y_train)

# In[172]:

svm_predictions = rbf_svm.predict(X_test)

```

Run: sentiment_classification

"C:\Program Files\Python39\python.exe" "C:/Users/keert/Desktop/Sudharsan/Learnings/Project materials/Audio-Sentiment-Analysis-master/Audio-Sentiment-Analysis-m

Accuracy using SVM Model: 0.44212962962962965
 Accuracy using RandomForest Model: 0.5185185185185185
 Accuracy using KNN Model: 0.45601851851851855

7. SVM- kernel= “poly”; degree=12; c=5000
8. Random forest n_estimators=70, criterion='entropy', random_state=42
9. KNN, nearest neighbour=1

```

75 ##### MODEL 2 #####
76
77
78 In[192]:
79
80
81 X_train, X_test, y_train, y_test = train_test_split(feature_all, y, test_size=0.3, random_state=2)
82
83
84
85
86 classifier = RandomForestClassifier(n_estimators=70, criterion='entropy', random_state=42)
87 classifier.fit(X_train, y_train)
88
89

```

Run: sentiment_classification

```

"C:\Program Files\Python39\python.exe" "C:/Users/keert/Desktop/Sudharsan/Learnings/Project materials/Audio-Sentiment-Analysis-master/Audio-Sentiment-Analysis-m
Accuracy using SVM Model: 0.4791666666666667
Accuracy using RandomForest Model: 0.5740740740741
Accuracy using KNN Model: 0.5416666666666667

Process finished with exit code 0

```

You are using code cells in the editor
PyCharm Professional Edition has special support for it.

PEP 8: E303 too many blank lines (4)

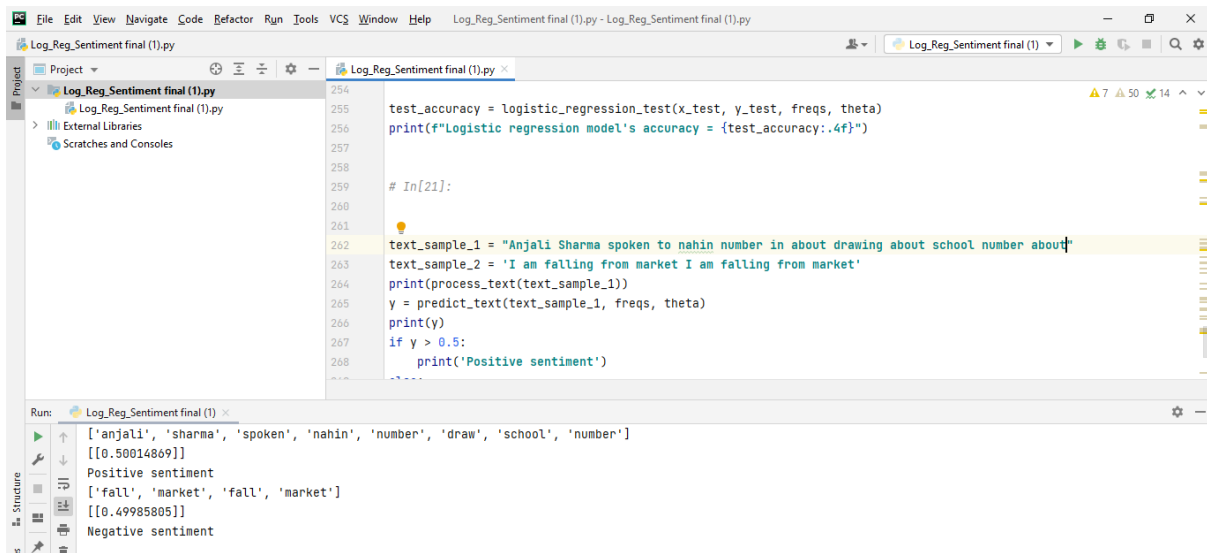
6:46 AM 12/14/2021

4.4 Classification based on Text

Given the test data and the weights of trained model, we calculated the accuracy of our logistic regression model.

- Using predict_tweet() function to make predictions on each tweet in the test set.
- If the prediction is > 0.5 , set the model's classification y_{hat} to 1, otherwise set the model's classification y_{hat} to 0.
- A prediction is accurate when y_{hat} equals test_y. Sum up all the instances when they are equal and divide by m.

Output



The screenshot displays a Jupyter Notebook environment. The main area shows a Python script for testing a logistic regression model. The script defines a function `logistic_regression_test` and uses it to evaluate the model's performance on two text samples. The first sample is "Anjali Sharma spoken to nahin number in about drawing about school number about" and the second is "I am falling from market I am falling from market". The script prints the accuracy of the model and the predicted sentiment for each sample. The output shows an accuracy of 0.50014869 for the first sample and 0.49985805 for the second sample. The sentiment is predicted as "Positive sentiment" for the first sample and "Negative sentiment" for the second sample.

```
254 test_accuracy = logistic_regression_test(x_test, y_test, freqs, theta)
255 print(f"Logistic regression model's accuracy = {test_accuracy:.4f}")
256
257
258 # In[21]:
259
260
261
262 text_sample_1 = "Anjali Sharma spoken to nahin number in about drawing about school number about"
263 text_sample_2 = "I am falling from market I am falling from market"
264 print(process_text(text_sample_1))
265 y = predict_text(text_sample_1, freqs, theta)
266 print(y)
267 if y > 0.5:
268     print('Positive sentiment')
```

Run: Log_Reg_Sentiment final (1) ×

```
['anjali', 'sharma', 'spoken', 'nahin', 'number', 'draw', 'school', 'number']
[[0.50014869]]
Positive sentiment
['fall', 'market', 'fall', 'market']
[[0.49985805]]
Negative sentiment
```

Accuracy for Logistic regression= 0.9834

Output for chunk 09 and chunk 06

['anjali', 'sharma', 'spoken', 'nahin', 'number', 'draw', 'school', 'number']

[[0.50014869]]

Positive sentiment

['fall', 'market', 'fall', 'market']

[[0.49985805]]

Negative sentiment

Chapter 5: FUTURE WORKS

- A model is to be developed to get the weighted average of the accuracies calculated from both the classifications of audio chunks and texts derived from the audio which is now limited.
- Accuracy of audio classification must be improved as the major part emotion will be displayed in the pitch of the customer.
- Audio can be further divided into smaller chunks where the conversation of customers can be divided into smaller parts to run through reinforcement learning and train the initial part to test the latter parts.
- In this way pitch of the person can be identified as pitch is a personal trait and louder mouth can be classified and not directly noted as anger person.
- This will help the model to improve the accuracy invariably.
- This will be stepping stone for machines to take over customer care which will reduce the man power in the industry.

REFERENCES

- [1] Cowie R, Douglas-Cowie E, Tsapatsoulis N, Votsis G, Kollias S, Fellenz W, Taylor JG. Emotion recognition in human-computer interaction. *IEEE Signal Process Mag* 2001;18(1):32–80.
- [2] Gunawardana A, Mahajan M, Acero A, Platt JC. Hidden conditional random fields for phone classification. In: Ninth European Conference on Speech Communication and Technology.
- [3] Wang SB, Quattoni A, Morency L-P, Demirdjian D, Darrell T. Hidden conditional random fields for gesture recognition. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2. IEEE; 2006. p. 1521–7.
- [4] Quattoni A, Wang S, Morency L-P, Collins M, Darrell T. Hidden conditional random fields. *IEEE Trans Pattern Anal Mach Intell* 2007;10:1848–52.
- [5] Farzaneh-Gord M, Mohseni-Gharyehsafa B, Arabkoohsar A, Ahmadi MH, Sheremet MA. Precise prediction of biogas thermodynamic properties by using ann algorithm. *Renewable Energy* 2020;147:179–91.
- [6] Ramezanizadeh M, Ahmadi MH, Nazari MA, Sadeghzadeh M, Chen L. A review on the utilized machine learning approaches for modeling the dynamic viscosity of nanofluids. *Renew Sustain Energy Rev* 2019;114:109345.
- [7] Kahani M, Ahmadi MH, Tatar A, Sadeghzadeh M. Development of multilayer perceptron artificial neural network (mlp-ann) and least square support vector machine (lssvm) models to predict nusselt number and pressure drop of tio₂/ water nanofluid flows through non-straight pathways. *Numer Heat Transfer, Part A: Appl* 2018;74(4):1190–206.
- [8] Baghban A, Kahani M, Nazari MA, Ahmadi MH, Yan W-M. Sensitivity analysis and application of machine learning methods to predict the heat transfer performance of cnt/water nanofluid flows through coils. *Int J Heat Mass Transf* 2019;128:825–35.
- [9] Ratnaparkhi A. A maximum entropy model for part-of-speech tagging. In: Conference on Empirical Methods in Natural Language Processing.

- [10] McCallum A, Freitag D, Pereira FC. Maximum entropy markov models for information extraction and segmentation. *ICML 2000*;17:591–8.
- [11] Kuo H-KJ, Gao Y. Maximum entropy direct models for speech recognition. *IEEE Trans Audio, Speech, Language Process* 2006;14(3):873–81.
- [12] Muhammad Hameed Siddiqi, An improved gaussian mixture hidden conditional random fields model for audio-based emotions classification, *Egyptian Informatics Journal* 22 (2021) 45–51
- [13] R Neumayer , A Rauber. Integration of Text and Audio Features for Genre Classification in Music Information Retrieval. In: *Advances in Information Retrieval*; 2007. p. 724-727.
- [14] Y Yang, Y Lin, H Cheng, I Liao, Y Ho, H Chen. Toward Multi-Modal Music Emotion Classification. *Advances in Multimedia Information Processing- PCM*; 2008. p. 70-79.
- [15] X Hu, J Downie, A Ehmman. Lyric Text Mining in Music Mood Classification. 10th International Symposium on Music Information Retrieval, Kobe; 2009 . p. 411-416.
- [16] G Mishne. Experiments with Mood Classification in Blog Posts. In : *Proceedings of ACM SIGIR, 2005 Workshop on Stylistic Analysis of Text for Information Access*; 2005.
- [17] Souraya Ezzat, Neamat El Gayar, Moustafa M Ghanem. Sentiment Analysis of Call Centre Audio Conversations using Text Classification, *International Journal of Computer Information Systems and Industrial Management Applications*, 2012;4:619-27.
- [18] Shivhare, S.N., Khethawat, S.. Emotion detection from text. *arXiv preprint arXiv:12054944* 2012;.
- [19] Sutoyo, R., Chowanda, A., Kurniati, A., Wongso, R.. Designing an emotionally realistic chatbot framework to enhance its believability with aiml and information states. *Procedia Computer Science* 2019;157:621–628.
- [20] Hajar, M., et al. Using youtube comments for text-based emotion recognition. *Procedia Computer Science* 2016;83:292–299.
- [21] Banik, N., Rahman, M.H.H.. Evaluation of naïve bayes and support vector machines on bangla textual movie reviews. In: *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*. IEEE; 2018, p. 1–6.

- [22] Hasan, M., Rundensteiner, E., Agu, E.. Automatic emotion detection in text streams by analyzing twitter data. *International Journal of Data Science and Analytics* 2019;7(1):35–51.
- [23] Liu, Z.T., Wu, M., Cao, W.H., Mao, J.W., Xu, J.P., Tan, G.Z.. Speech emotion recognition based on feature selection and extreme learning machine decision tree. *Neurocomputing* 2018;273:271–280.