
Threatening Calls Alert System

Contents

1	Introduction	2
2	Materials & Methods	2
2.1	Data Collection	2
2.2	Recording and Converting Audio to Text	2
2.3	Cleaning the Dataset	2
2.4	Loading and Encoding the Data	2
2.5	Training the Data with Different ML Models	3
3	Results and Performance	3
4	Automatic Threatening Call Alert System Implementation	4
5	Conclusion	4

1 Introduction

The Automatic Threatening Call Alert System is developed to address the global concern of threatening calls, which cause psychological distress and disruptions in daily life for individuals and communities. Using advanced machine learning algorithms, the system accurately classifies threatening calls. A labeled dataset, comprising threatening and non-threatening call examples, was created manually for training and testing the models. The system takes audio input, converts it to text using speech recognition, and then classifies the call as threatening (labeled 0) or non-threatening (labeled 1). Its major goal is to quickly warn users via WhatsApp messaging when a potential threat is detected, thereby reducing the impact of such calls on users well-being and promoting a safer environment.

2 Materials & Methods

2.1 Data Collection

A labeled dataset, comprising threatening(labeled 0) and non-threatening(labeled 1) call was created manually for training and testing the models samples.

2.2 Recording and Converting Audio to Text

The system captures audio from the microphone and converts it into text using pyaudio and speech recognition libraries.

2.3 Cleaning the Dataset

The dataset was loaded and cleaned by removing NaN values and stop-words from the sentences using the NLTK library.

2.4 Loading and Encoding the Data

The cleaned dataset was loaded, and the text data was encoded using the CountVectorizer to convert it into bag-of-words features.

2.5 Training the Data with Different ML Models

The data was split into training and testing sets. Four different machine learning models were trained on the training data:

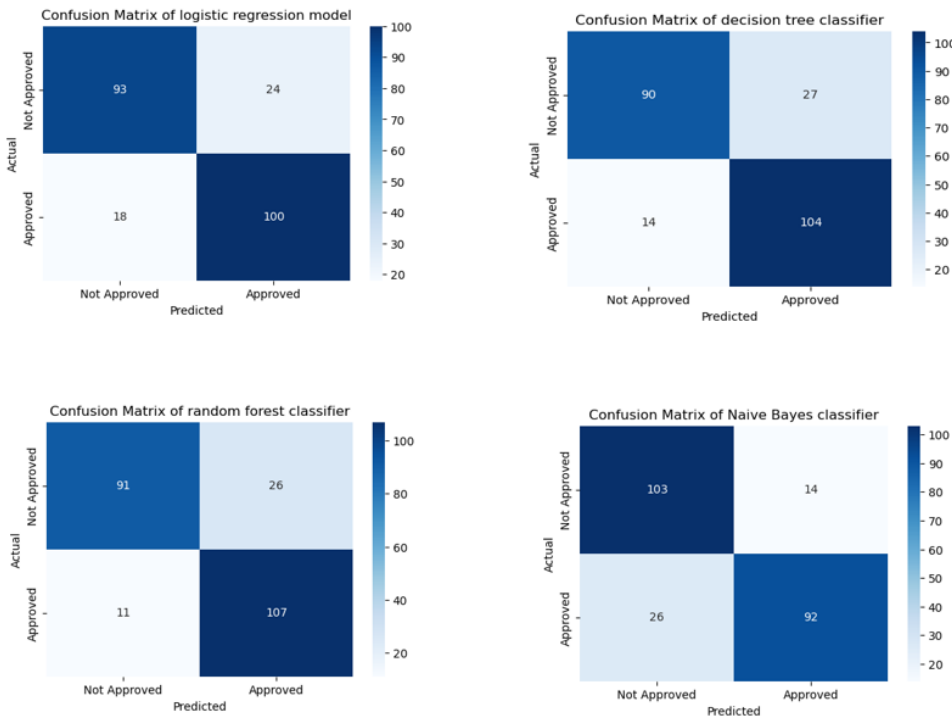
- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- Naive Bayes Classifier

3 Results and Performance

The performance of the trained models on the test set was as follows:

S.no	Model Name	Accuracy
1	Logistic Regression	82.13%
2	Decision Tree Classifier	82.55%
3	Random Forest Classifier	84.26%
4	Naive Bayes Classifier	82.98%

The confusion matrices for the four models:



The Random Forest Classifier excelled in the evaluation, achieving the highest accuracy among tested algorithms. Its success can be attributed to handling complex datasets, avoiding overfitting, and aggregating predictions from multiple trees for improved robustness. The higher accuracy indicates effective capture of patterns in threatening and non-threatening call data, making it invaluable for the Automatic Threatening Call Alert System.

4 Automatic Threatening Call Alert System Implementation

The text generated from the audio was tested using the Random forest model to determine whether or not the call is threatening. To construct the Automatic Threatening Call Alert System, the model was connected with WhatsApp using the pywhatkit library. When a threatening call is detected, the system immediately sends alert messages to the selected recipients via WhatsApp. This quick notification enables users to take the required safeguards and seek assistance as soon as possible.

5 Conclusion

The Automatic Threatening Call Alert System provides an effective and proactive solution to combat threatening calls. By leveraging machine learning and WhatsApp messaging, the system accurately identifies potential threats, ensuring timely alerts to protect individuals and communities from distressing experiences.

References

1. PyAudio. (n.d.). Retrieved from <https://people.csail.mit.edu/hubert/pyaudio/>
2. SpeechRecognition. (n.d.). Retrieved from https://github.com/Uberi/speech_recognition
3. scikit-learn. (n.d.). Retrieved from <https://scikit-learn.org/>
4. PyWhatKit. (n.d.). Retrieved from <https://github.com/Ankit404butfound/PyWhatKit>