**Outlining Document**

**Project by**
**Prudhvi Raj Maharana**
**Rajendera Kumar Behara**
**Syed Ali Hussain**

1. Who is your stakeholder?

The stakeholder is a research team focused on environmental sound classification. They are interested in developing a robust machine learning model to classify various environmental sounds accurately.

2. What is the problem they are trying to solve?
The problem is to develop an accurate and efficient model for classifying environmental sounds into predefined categories. This can be used for applications such as urban sound monitoring, wildlife monitoring, and noise pollution analysis.

3. Where is your dataset from?
The dataset used is the ESC-50 dataset, which is a labeled collection of 2000 environmental audio recordings organized into 50 classes. The dataset can be accessed from the following link: ESC-50 Dataset.

4. What models did you try, and why did you choose those models?

k-Nearest Neighbors (k-NN)

- **Reason for Choice:** k-NN is a simple, instance-based learning algorithm that is easy to implement and understand. It is often used as a baseline model for classification tasks.
- **Pros:** Simple to implement, no training phase, effective for small datasets.
- **Cons:** Computationally expensive during prediction, sensitive to irrelevant features and the choice of k.

Random Forest (RF)

- **Reason for Choice:** Random Forest is an ensemble learning method that constructs multiple decision trees and aggregates their results. It is robust to overfitting and can handle high-dimensional data well.
- **Pros:** Handles large datasets well, reduces overfitting, provides feature importance.
- **Cons:** Can be computationally intensive, less interpretable than single decision trees.

Support Vector Machine (SVM)

- **Reason for Choice:** SVM is a powerful classifier that works well with high-dimensional data and is effective in cases where the number of dimensions exceeds the number of samples.

- **Pros:** Effective in high-dimensional spaces, robust to overfitting.
- **Cons:** Computationally intensive, sensitive to the choice of kernel and hyperparameters.

Gradient Boosting

- **Reason for Choice:** Gradient Boosting is a powerful ensemble learning technique that combines multiple weak learners, typically decision trees, to create a strong predictive model.
- **Pros:** Feature importance, flexibility and ability to handle non-linear relationships.

## 5. What features did you select/engineer? How did you choose those?

Mel-frequency Cepstral Coefficients (MFCCs)

- **Reason for Choice:** MFCCs are widely used in audio processing as they capture the timbral/textural characteristics of the sound, which are important for distinguishing different environmental sounds.
- **Selection Process:** Computed using the librosa library, 13 MFCC coefficients were derived from the log Mel-spectrograms.

Zero-Crossing Rate (ZCR)

- **Reason for Choice:** ZCR is a simple measure of the noisiness of the signal, which can help differentiate between different types of sounds.
- **Selection Process:** Computed frame-wise by counting the number of times the signal changes sign per frame.

## 6. How did you evaluate the model? What evaluation metrics did you use? Why?

Evaluation Metrics

- **Accuracy:** Measures the proportion of correctly classified instances out of the total instances. It is a straightforward metric for classification tasks.
- **Confusion Matrix:** Provides a detailed breakdown of the model's performance by showing the true positives, false positives, true negatives, and false negatives for each class.
- **Precision and Recall:** Precision measures the proportion of true positive predictions out of all positive predictions, while recall measures the proportion of true positive predictions out of all actual positives. These metrics are important for understanding the model's performance in imbalanced datasets.

Evaluation Process

The models were evaluated using a 5-fold cross-validation approach to ensure that the results are robust and not dependent on a specific train-test split. The confusion matrix was used to visualize the performance of each classifier.

## 7. What would you do differently next time or given more time, what would your future work be?

Given more time, the following improvements could be made:

- **Hyperparameter Tuning:** Perform a more extensive hyperparameter search to optimize the performance of the models.
- **Feature Engineering:** Explore additional features such as spectral contrast, chroma features, and others to improve classification accuracy.
- **Advanced Models:** Experiment with deep learning models such as Convolutional Neural Networks (CNNs) that have shown promise in audio classification tasks.
- **Data Augmentation:** Use data augmentation techniques to increase the diversity of the training data and improve model generalization.

8. Do you recommend your client use this model? Is the precision/recall good enough for the intended use case?

Based on the results, the Random Forest classifier performed the best among the models tried, with an accuracy of approximately 74.5%. However, this is still below human performance (81.3%). The precision and recall for the Random Forest model are reasonable, but there is room for improvement. For the intended use case, it is recommended to use the Random Forest model as a baseline and explore further improvements as outlined in the future work section.

9. How will you deploy your model?

The model can be deployed as a web service using frameworks such as Flask or FastAPI. The deployment process would involve:

1. **Model Serialization:** Save the trained model using a library like joblib or pickle.
2. **API Development:** Develop an API that accepts audio files as input, processes them to extract features, and returns the predicted class.
3. **Containerization:** Use Docker to containerize the application for easy deployment and scalability.
4. **Hosting:** Deploy the containerized application on a cloud platform such as AWS, Google Cloud, or Azure.

Link to video presentation: https://youtu.be/7YgtS_Fg6fA
Github link to code and ppt: https://github.com/SyedAliHussain0898/Final-Project