Gender Recognition Using Voice

Amey Admane[1], Neha Pandey[1], Pratik Yawalkar[1], Vaidehi Choudhari[1], Sweta Jain[1]
[1]Student, Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, admaneak@rknec.edu
[1]Student, Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, pandeynj@rknec.edu
[1]Student, Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, yawalkarpy@rknec.edu
[1]Student, Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, choudharivd@rknec.edu
[2]Assistant Professor, Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra,jains@rknec.edu

***Abstract*—Gender Recognition using voice will be of huge importance in the near future technology as its uses could range from smart assistance(robots) to customer service sector and many more. Machine learning models play a vital role in achieving this task. Using the acoustic properties of voice, different ML models will classify the gender as male and female.In this project we have used the following models- Random Forest, Decision Tree, Logistic Regression, Support Vector Machine (SVM), Gradient Boosting, K-Nearest Neighbor (KNN), and ensemble method (KNN, logistic regression, SVM). To propose which algorithm is best for recognizing gender, we have evaluated the models based on results obtained from analysis of accuracy, recall, F1 score, and precision.**

Keywords—Gender recognition, random forest, decision tree, logistic regression, gradient boosting, KNN, SVM, ensemble (KNN, logistic regression, and SVM), best model forgender recognition

I. INTRODUCTION

In this project, we aim to identify the gender using voice by training the Kaggle[1] dataset using different machine learning algorithms and suggesting the best machine learningalgorithm to accomplish the task. The Machine learning algorithms that we have implemented are Decision Tree,Logistic Regression, K Nearest Neighbors, Support Vector Machine, Gradient Boosting, Random Forest, and Ensemble model built using KNN, SVM, and Logistic Regression. The results of these models are scrutinize using six different comparison parameters: Precision, Recall, F1-score, Accuracy, ROC curve, and Confusion matrix. The model with the highest score will be the best model to accomplish the task of gender recognition using voice.

The future purview of our project ranges from being usefulin the customer service sector to high technological advances. There are many applications where gender recognition can be useful. It is used in personal assistants, automatic salutations, to detect the gender of criminals using voice, and many other. In countries like Japan, China, and Korea there are fully automated stores with no human employees and many other countries are planning to build these stores in the future. So, in these automated shops predicting gender by voice will help to recommend products based on gender.

II. RELATED WORK

In the past few years, using machine learning models a lotof drudgery had already been done in the field of gender identification by voice. These machine learning models make use of different acoustic properties like pitch, frequency, kurtosis, peak frequency, and many other properties to predict gender.

For gender recognition by voice Ioannis E. Livieris, Emmanuel Pintelas and Panagiotis Pintelas proposed an ensemble-based self-labeled algorithm built using the three most efficient self-labeled methods: Tri-training, Co- training, and Self-training by using the ensemble as a base learner, which they termed as iCST-Voting. For evaluating the performance, they used the Kaggle dataset and the Deterding dataset. They got the highest accuracy of 98.42%[2].

In a study conducted by Mucahit Buyukyilmaz and Ali Osman Cibikdiken, they used a Multilayer perceptron deep learning model to concede gender by voice. On testing it on the Kaggle dataset they got an accuracy of 96.74%[3].

For recognizing the gender using voice Zvarevashe et al. performed feature selection through the random forest recursive feature elimination with gradient boosting machines. Without feature selection, the gradient boosting algorithm attained an accuracy of 97.58% and with feature selection, it reached an accuracy of almost 100%[4].

To solve the problem of gender recognition by voice, Harb and Chen used a set of gaussian modeling features, pitch related features, and contrasting acoustic propertieswith dissimilar classifiers. They used a neural network for constructing the classifier[5].

III. DATASET

Our project aims to identify gender based on voice. For this, we have taken our dataset from Kaggle. It consists of 3168 samples of recorded voice of which 1584 are males and 1584 are females as shown in Figure 1.

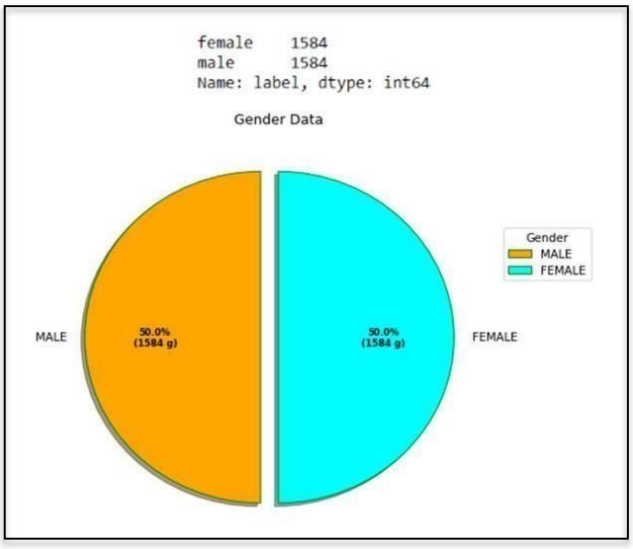

Figure 1: Distribution of Data

IV. METHODOLOGY

Figure 2 represents the workflow of the project for achieving the end goal of recognizing the gender using voice.

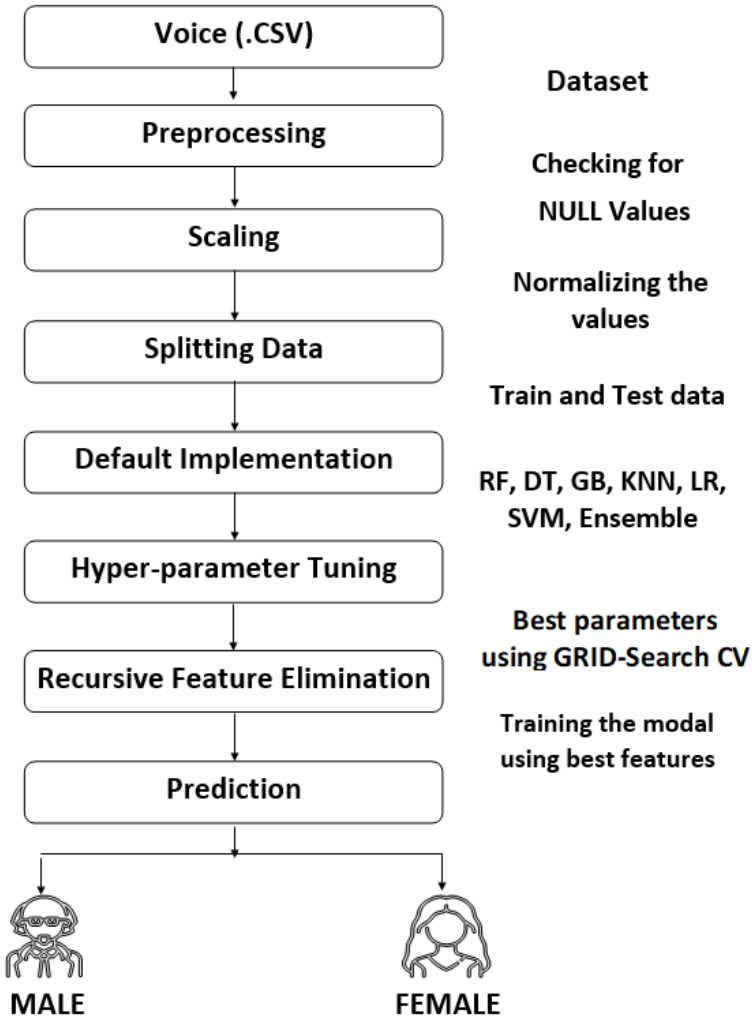

Figure 2: Workflow of project

A. Pre-Processing of Dataset

First, we have pre-processed dataset taken from Kaggle. This helps in assuring the maximum accuracyof the models which are later applied to this dataset. The steps followed to achieve our goal are listed below:

i. Checking for NULL values

One of the most important steps in data preprocessing is to check for the null values. It is important because presence of any null value can directly degrade the performance of any machine l earning model. So, after checking the dataset for null values, we found that our dataset consisted of no null values.

ii. Checking for Numeric and Non-Numeric Features

It is important to check the categorical values and convert them to numerical values because any machine learning model can only processnumerical values. On checking, we found that of all the 21 features only the label was the feature that consisted of non-numerical values, the rest of them were numerical values. So, In the label column, we represented males using 1 and females using 0.

iii. Checking for Dependent and Independent Features

Success of any machine learning model depends on identifying correct dependent and independent variables. On finding this, we found that our datasetconsisted of 20 independent features and 1 dependent feature 'label'.

B. Scaling

While training a few of the models it is necessary to doscaling. Machine learning algorithms only deal with numbers and have no knowledge about what these numbers represent. So, it may happen that numbers having higher values will be given more importance than other numbers, which can directly affect our output and degrade performance. So, in a few of the models where such a situation can arise, we have done scaling.

C. Splitting Training and Testing data

Before training, for avoiding overfitting of data and to see how the model will perform with the new data we have split the dataset into training and testing datasets. We divided the dataset into two parts in a 2:8 ratio i.e., 20% of the dataset for testing and 80% of the dataset for training.

D. Default Model

First, we have trained and tested the models using thedefault parameters of the algorithm using all 20 features. But this is not optimal and time-consuming and can also cause overfitting. Here, Hyperparameter tuning comes in handy as it helps in maximizing the performance of the model without causing the before stated issues.

E. Hyperparameter Tuning

To find the best parameters, we have done hyperparameter tuning using GridSearchCV for all the models. Then we trained our model on the hyperparameters that we got. In some algorithms, we got better accuracy after doing hyperparameter tuning.

F. Recursive Feature Elimination

While training any machine learning model it is notalways efficient to train themodel using all the features in dataset. Because certain features are irrelevant in predicting the target variable. So, we can drop thosefeatures. This will also help us to reduce the time and thecomputations required to train the model without affecting the performance of the model. So, using RFE from sklearn.feature_selection we can achieve this. We have done the recursive feature elimination for all the models.

G. Models

We have trained our model with the following algorithms.

i. Random Forest

Random forest is referred to as an ensemble machine learning algorithm. We have used the RandomForestClassifier from sklearn library to implement the model. We have not done scaling forrandom forest. We first trained the model using default parameters and then we did hyperparameter tuning.

ii. Support Vector Machine

Support Vector Machine can be applied for both regression as well as classification problems. It solves the problem by finding a hyperplane in an N-dimensional space that segregates the data points. Before training the model, scaling is performed by using the StandardScaler() method of the scikit learn library. In SVM there are many different types of kernel. Out of which we have implementedthree kernels: linear, RBF, and polynomial.

- SVM Linear Model

  Linear Kernel is found to be the elite choice when there is a wide range of features available. Compared to other kernelsin SVM, Linear kernel functions are comparatively faster. In the SVM-Linear kernel, the most important parameter is "C". We have taken a wide range of numbers starting with zero and got the best values for C as 0.014. The value of C can't be less than or equal to zero as in that case there would be difficulty in plotting the hyperplane (hard margin will be there).

- SVM Radial Bias Function Model

  RBF kernel is mostly chosen when data isnot linear. Even in cases where proper knowledge related to data is not present evenin those cases, it makes proper separation.

- SVM Polynomial Model

  The similarity in the training samples can be determined using a polynomial kernel. Tofind similarities, it looks at the features of the input samples and combination of it.

iii. Gradient Boosting

Initially, we trained the model using the GradientBoostingClassifier module of Python's Scikit Learn library. Then we applied the Grid Search CV method on n_estimators, learning_rate and max_depth parameters for a bunch of values andfound out the best values for the model to be able togive the most accurate results. Next, we found out the most related features using recursive feature elimination.

iv. Decision Tree

We made use of ID3 algorithm for implementing the Decision Tree. It operates by choosing the features with maximum information gain. We have calculated the information gain using both entropy and Gini index criteria with the help of the DecisionTreeClassifier module of Python's Scikit Learn library. The formula used for calculating entropy and Gini index is-

Entropy $= \sum_{i=1}^{c} -fi$ Equation (1)

Gini $= \sum_{i=1}^{c} fi(1 - fi)$ Equation (2)

Where fi is the frequency of label i at a node and c is the number of unique labels.

v. K-Nearest Neighbors

First, we implemented the KNN algorithm ondefault parameters using the KNeighborClassifier module from scikit learn library. After that Grid Search CV was performed on n_neighbors, weights, metric parameters, and parameters value giving the best score were figured out. Result values were calculated for these hyperparameter values.

vi. Logistic Regression

We have implemented logistic regression using the sklearn linear model. There are different parameters in logistic regression like solvers, penalty, C, max_iter, and many others. Out of this, we have set the solver (algorithm used for optimization) to lgbs and penalty(type of regularization) to L2 because we want to shrink the coefficients without making them zero. For other parameters, we have used hyperparameter tuning.

vii. Ensemble

This is an ensemble model which will be using three different classifiers KNN, SVM, and Logistic regression to predict the final output. This will help us to improve the accuracy. All these three classifiers will be predicting the output, then using mode final output will be predicted.

V. RESULTS

Comparison between three SVM kernel models: SVM linear, SVM radial bias function, and SVM polynomial model is depicted in Figure 3 based on accuracy, precision, F1- score, and recall.

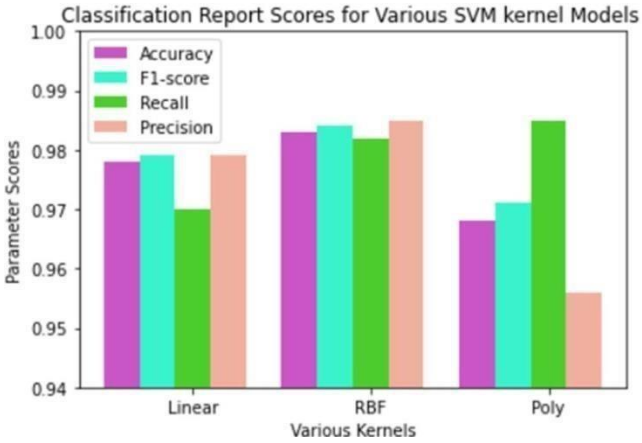

Figure 3: Comparison of SVM Kernels

The results obtained from all the models before and after applying hyperparameter tuning are summarized in Table 1 as shown below.

Table 1: Result of Hyperparameter Tuning

Models		Accuracy	Precision	Recall	F1 Score	AUC
Decision Tree	Before	0.9511	0.9543	0.9451	0.9497	0.9509
	After	0.9605	0.9765	0.9419	0.9589	0.9601
Random Forest	Before	0.9747	0.9803	0.9677	0.9740	0.9746
	After	0.9668	0.9737	0.9580	0.9658	0.9666
Gradient Boosting	Before	0.9700	0.9739	0.9645	0.9692	0.9699
	After	0.9731	0.9771	0.9677	0.9724	0.9730
Support Vector Machine	Before	0.9826	0.9851	0.9821	0.9836	0.9826
	After	0.9779	0.9764	0.9821	0.9792	0.9776
K Nearest Neighbours	Before	0.9794	0.9792	0.9824	0.9807	0.9793
	After	0.9810	0.9821	0.9821	0.9835	0.9793
Logistic Regression	Before	0.9810	0.9850	0.9792	0.9821	0.9811
	After	0.9810	0.9850	0.9792	0.9821	0.9813
Ensemble Model	Before	0.9873	0.9910	0.9851	0.9880	0.9875
	After	NA	NA	NA	NA	NA

From Table 2, It can be observed that the Ensemble algorithm built using KNN, SVM and logistic regression performed the best.

Figure 4 and Figure 5 shows the confusion matrix and ROC curve for the Ensemble Model.

.

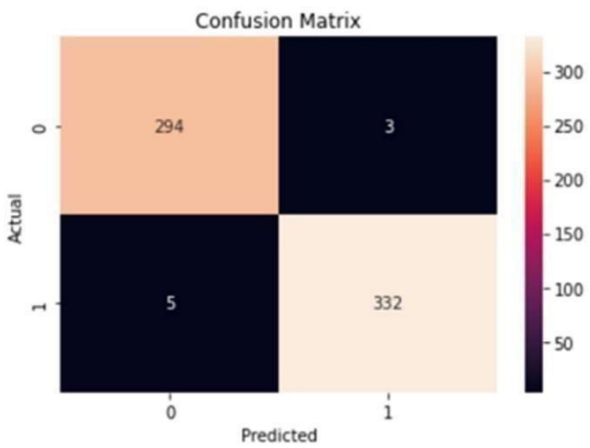

Figure 4: EM Confusion Matrix

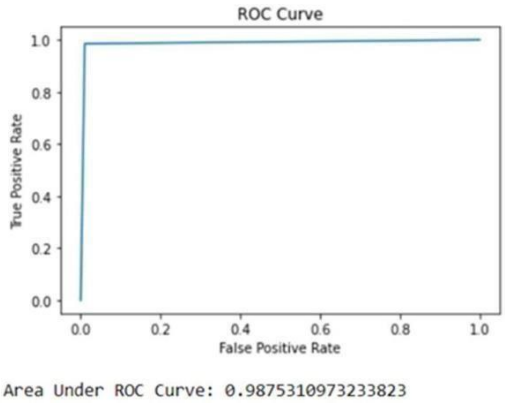

Figure 5: EM ROC Curve

For some models, results were better after performing hyperparameter tuning and for others default, algorithm results were better. After performing recursive elimination, we were able to optimize the time required to train the modelas we were using the best 10 features to train the model instead of using all 20 features. The accuracy and time taken are summarized in Table 3.

Table 3: Accuracy and Time taken by models

Model	Default Model	Time	Grid Search CV	Time	Recursive Feature Elimination	Time
Decision Tree	0.9511	0.0568	0.9542	0.0757	0.9558	0.0264
Random Forest	0.9747	0.7393	0.9652	13.2550	0.9700	0.6579
Gradient Boosting	0.9700	0.8651	0.9731	4.3757	0.9716	2.407
Support Vector Machine	0.9826	0.0765	0.9779	0.0534	NA	NA
K Nearest Neighbours	0.9794	0.1148	0.9794	0.0294	NA	NA
Logistic Regression	0.9810	0.4909	0.9810	0.0672	0.9810	0.0234
Ensemble	0.9873	0.2106	NA	NA	NA	NA

Table 4. shows the best values for accuracy, precision, f1 and recall scores of all the models after performing GridSearchCv and recursive feature elimination.

Table 4: Result of all Models

Model	True Negative	False Negative	True Positive	False Positive	Area Under Curve
Decision Tree	315	20	290	9	0.9538
Random Forest	318	10	300	6	0.9746
Gradient Boosting	317	10	300	7	0.9730
Logistic Regression	292	7	330	5	0.9811
Ensemble	294	5	332	3	0.9875
K Nearest Neighbours	290	6	331	7	0.9793
Support Vector Machine	293	7	330	4	0.9828

From Table 4, We can observe that the overall performance of the ensemble algorithm built using SVM, LR, and KNN is best followed by SVM. Generally, Ensemble algorithms perform better than other models as they are built using a number of classifiers but in our case SVM performed better than random forest and gradient boosting may be because of the use of weak classifiers.

The bar chart in Figure 6 represents the comparison of models based on accuracy, precision, F1 score, and recall.

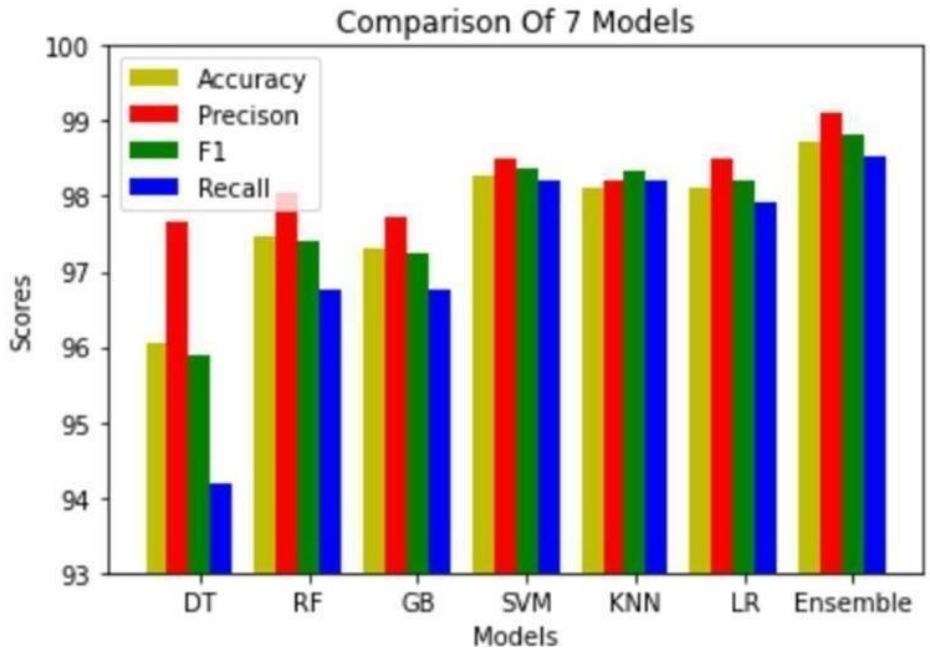

Figure 6: Comparison of 7 Models

VI. CONCLUSION

We successfully trained the Kaggle dataset using differentmachine learning algorithms. Based on the accuracy, precision, recall, F1-score, Roc curve, and confusion matrix we found that the overall performance of ensemble model build using KNN, SVM, and logistic regression were best both in terms of time and accuracy. It gave us the highest accuracy of 98.731% which was followed by SVM.

REFERENCES

[1] https://www.kaggle.com/primaryobjects/voicegender

[2] Livieris, Ioannis & Pintelas, Emmanuel & Pintelas, P.. (2019). **Gender Recognition by Voice using an Improved Self-Labeled Algorithm**. Machine Learning and KnowledgeExtraction. 1. 492-503. 10.3390/make1010030.

[3] Büyükyılmaz, Mücahit & Çıbıkdiken, Ali. (2016). **Voice GenderRecognition Using Deep Learning**.10.2991/msota-16.2016.90.

[4] Zvarevashe, K.; Olugbara, O.O. **Gender Voice Recognition Using Random Forest Recursive Feature Elimination with Gradient Boosting Machines**. In Proceedings of the 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD), Durban, SouthAfrica, 6–7 August 2018; pp. 1–6.

[5] Harb, H.; Chen, L. **A general audio classifier based on human perception motivated model**. Multimed. Tools Appl. 2007, 34,375–395.