GENDER VOICE RECOGNITION USING MACHINE LEARNING

A project report submitted in partial fulfillment
of the requirements for the award of degree in

Master of Computer Application (MCA)

By

Tadi Mukesh
(Regd. No: 121722501022)

Under the esteemed guidance of

Vagolu S Prasad Babu
Assistant Professor

Department of Computer Science
GITAM Institute of Science
GITAM (Deemed to be University)
Visakhapatnam -530045, A.P
(2020)

CERTIFICATE

This is to certify that the project entitled **"Gender Voice Recognition Using Machine Learning"** is a bonafide work done by **Tadi Mukesh**, Regd. No: 121722501022 during **December 2019** to **April 2020** in partial fulfillment of the requirement for the award of degree of **Master of Computer Application (MCA)** in the Department of Computer Science, GITAM Institute of Science, GITAM (Deemed to be University).

Internal Guide
Mr. Vagolu S Prasad Babu
Assistant Profesor

Head of the Department
Prof. K. Vedavathi
HOD

External Examiner

DECLARATION

I **Tadi Mukesh,** hereby declare that the project entitled **" Gender Voice Recognition Using Machine Learning"** is an original work done in the partial fulfillment of the requirements for the award of degree of **Master of Computer Application (MCA)** in **GITAM Institute of Science, GITAM (Deemed to be UNIVERSITY).** I assure that this project work has not been submitted towards any other degree or diploma in any other colleges or universities.

Tadi Mukesh
(Regd. No: 121722501022)

ACKNOWLEDGEMENT

It is my prime duty to express my sincere gratitude to all those who have helped me to successfully complete this project.

I express respectful and sincere thanks to my guide **Mr. Vagolu S Prasad Babu**, Assistant Professor, our H.O.D **Prof. K. Vedavathi**.

I express my gratitude to the faculty members of our department for the valuable cooperation, guidance and continuous support rendered by them on me throughout my major project.

At last I would like to thank my parents for giving needful advices and giving full support for completion of this major project

Tadi Mukesh
(Regd. No: 121722501022)

ABSTRACT

Gender identification is one of the major problem speech analysis today. Tracing the gender from acoustic data i.e., pitch, median, frequency etc. Machine learning gives promising results for classification problem in all the research domains. There are several performance metrics to evaluate algorithms of an area. Our Comparative model algorithm for evaluating different machine learning algorithms based on different metrics in gender classification from acoustic data. Agenda is to identify gender. The main parameter in evaluating any algorithms is its performance. Misclassification rate must be less in classification problems, which says that the accuracy rate must be high. Location and gender of the person have become very crucial in economic markets in the form of AdSense. Here with this comparative model algorithm, we are trying to assess the different ML algorithms and find the best fit for gender classification of acoustic data.

The dataset is pre-processed i.e. incomplete, inconsistent data is handled and the feature engineering is done i.e. finding most meaningful attribute for the problem statement. Finally, to identify a voice as male or female, based upon acoustic properties of the voice and speech. The dataset consists of 3,168 recorded voice samples, collected from male and female speakers

INDEX

1. Introduction:

1.1 About the Project:

Determining a person's gender as male or female, based upon a sample of their voice seems to initially be an easy task. In the real world, the human ear can easily detect the difference between a male or female voice within the first few spoken words. It is one of the most common means of communication in the world is through voice. Voice is filled with lots of linguistic features. These voice features are considered as the voice prints to recognize the gender of a speaker. The recorded voice is considered as the input to the system, which then the system process to get voice features. However, designing a computer program to do this turns out to be a bit trickier.

Tracing the gender from acoustic data i.e., pitch, median, frequency etc. Machine learning gives promising results for classification problem in all the research domains. There are several performance metrics to evaluate algorithms of an area. Our Comparative model algorithm for evaluating different machine learning algorithms based on different metrics in gender classification from acoustic data. Agenda is to identify gender.

This article describes the design of a computer program to model acoustic analysis of voices and speech for determining gender. Examine the input and compare it with the trained model, carry out calculations based on the algorithm used and gives the matching output i.e. male or female.

So, the goal is to remove highly correlated attributes and to find only those features that will influence the predicting model the most. So that in a smaller number of features we can get the most accurate prediction using any machine learning classification algorithm. Less the number of features to consider for the prediction model less will be the training time and more will be the accuracy as all unrelated feature are removed and not considered. This will in turn also reduce the overfitting of the model.

1.2 Introduction to the Problem Domain

Proposed project is a Machine Learning classification problem which build a gender recognition system based on voice using dataset.

- First, the dataset is pre-processed i.e. incomplete, inconsistent data is handled.
- Second, the feature engineering is done i.e. finding the most meaningful attribute for the problem statement.
- Finally, algorithms are applied to the dataset to know whether which algorithm is giving accurate values.

If we use all the feature that can influence the prediction, it will increase the training time of the model and will make the model more complex. But not all features are of high importance.

Although using many features can help us get more accurate data, but still it will take a lot of time to predict the outcome.

1.3 Applications of the System:

Gender identification is one of the major problem speech analyses today. Tracing the gender from acoustic data i.e., pitch, median, frequency etc. Machine learning gives promising results for classification problem in all the research domains. There are several performance metrics to evaluate algorithms of an area. Our Comparative model algorithm for evaluating different machine learning algorithms based on different metrics in gender classification from acoustic data. Agenda is to identify gender.

The main parameter in evaluating any algorithms is its performance. Misclassification rate must be less in classification problems, which says that the accuracy rate must be high. Location and gender of the person have become very crucial in economic markets in the form of AdSense. Here with this comparative model algorithm, we are trying to assess the different ML algorithms and find the best fit for gender classification of acoustic data.

2. Machine Learning approach

2.1 Machine learning Process

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to use statistical techniques to give computers the ability to learn with data, improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. Data is being produced and stored continuously ("big data"), which is used in various fields:

science: genomics, astronomy, material science, particle accelerators, sensor networks: weather measurements, traffic etc.

people: social networks, blogs, mobile phones, purchases, bank transactions.

Data is not random, it contains structure that can be used to predict outcomes, or gain knowledge in some way.

Ex: patterns of Amazon purchases can be used to recommend items.

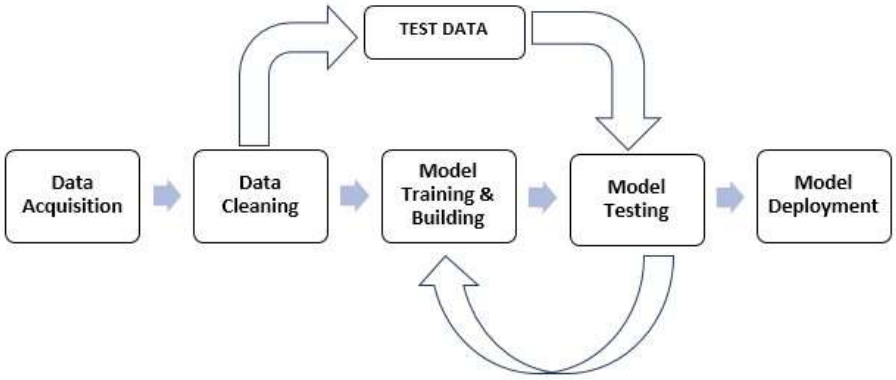


Fig 3. Machine Learning Process

Basic Machine Learning project follows these 7 steps to predict it output. The step are as follows:

1. **Gathering Data**
   Once you know exactly what you want and the equipment's are in hand, it takes you to the first real step of machine learning- Gathering Data. This step is very crucial as the quality and quantity of data gathered will directly determine how good the predictive model will turn out to be. The data collected is then tabulated and called as Training Data.

2. **Data Preparation**
   After the training data is gathered, you move on to the next step of machine learning: Data preparation, where the data is loaded into a suitable place and then prepared for use in

machine learning training. Here, the data is first put all together and then the order is randomized as the order of data should not affect what is learned.

This is also a good enough time to do any visualizations of the data, as that will help you see if there are any relevant relationships between the different variables, how you can take their advantage and as well as show you if there are any data imbalances present. Also, the data now must be split into two parts. The first part that is used in training our model, will be most of the dataset and the second will be used for the evaluation of the trained model's performance. The other forms of adjusting and manipulation like normalization, error correction, and more take place at this step.

3. **Choosing a model**
   The next step that follows in the workflow is choosing a model among the many that researchers and data scientists have created over the years. Make the choice of the right one that should get the job done.
4. **Training**
   After the before steps are completed, you then move onto what is often considered the bulk of machine learning called training where the data is used to incrementally improve the model's ability to predict. The training process involves initializing some random values for say A and B of our model, predict the output with those values, then compare it with the model's prediction and then adjust the values so that they match the predictions that were made previously. This process then repeats and each cycle of updating is called one training step.

5. **Evaluation**
   Once training is complete, you now check if it is good enough using this step. This is where that dataset you set aside earlier comes into play. Evaluation allows the testing of the model against data that has never been seen and used for training and is meant to be representative of how the model might perform when in the real world.

6. **Parameter Tuning**
   Once the evaluation is over, any further improvement in your training can be possible by tuning the parameters. There were a few parameters that were implicitly assumed when the training was done. Another parameter included is the learning rate that defines how far the line is shifted during each step, based on the information from the previous training step. These values all play a role in the accuracy of the training model, and how long the training will take.

7. **Prediction**
   Machine learning is basically using data to answer questions. So, this is the final step where you get to answer few questions. This is the point where the value of machine learning is realized. Here you can Finally use your model to predict the outcome of what you want.

2.2 Types of Machine Learning Algorithms

There are three types of Machine Learning Algorithms.

a) Supervised Learning
b) Unsupervised Learning
c) Reinforcement Learning

a) Supervised Machine Learning

Supervised learning algorithms are a type of Machine Learning algorithms that always have known outcomes. All supervised learning algorithms have a training phase (supervised means 'to guide'). The algorithm uses training data which is used for future predictions.

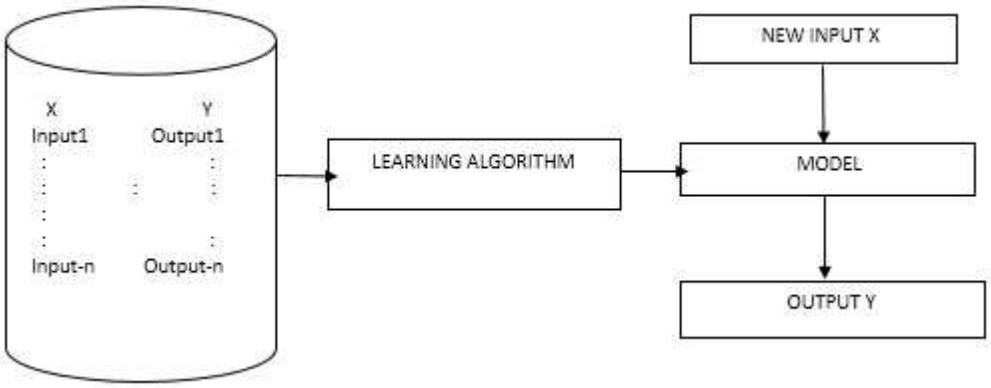


Fig 4. Working of Supervised Machine Learning

Supervised learning problems can be further divided into two parts, namely classification, and regression.

Classification: A classification problem is when the output variable is a category or a group, such as "black" or "white" or "spam" and "no spam".

Regression: A regression problem is when the output variable is a real value, such as "Rupees" or "height."

b) Unsupervised Machine Learning

In unsupervised learning, there is no target or outcome variable to predict or estimate. It is used for clustering population in different groups. Mathematically, unsupervised learning is when you only have input data (X) and no corresponding output variables. Unsupervised learning problems can be further divided into association and clustering problems. **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as "people that buy X also tend to buy Y".

Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

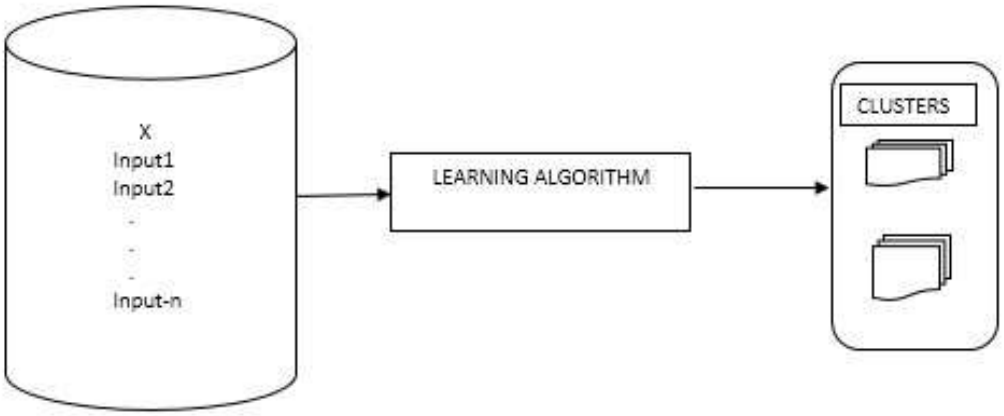


Fig 5. Working of Unsupervised Machine Learning

c) Reinforcement Learning

A computer program will interact with a dynamic environment in which it must perform a particular goal (such as playing a game with an opponent or driving a car). The program is provided feedback in terms of rewards and punishments as it navigates its problem space. Using this algorithm, the machine is trained to make specific decisions. The machine is exposed to an environment where it continuously trains itself using trial and error method.

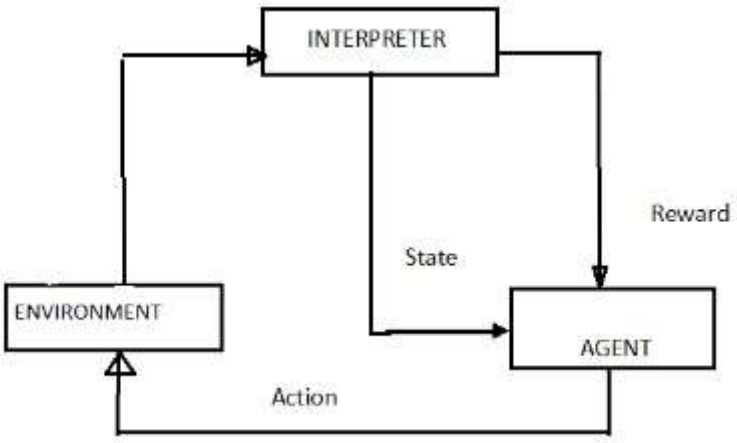


Fig 6. Working of Reinforcement Learning

3. System Requirements Specification:

3.1 Hardware Requirement

- System: Core i5 Processor 2.7GHz
- Hard Disk: 500MB
- RAM: 4GB
- Monitor: 15 VGA Colour.
- Input Device: Keyboard, Mouse

3.2 Software Requirement

- Operating system: Windows XP/7.
- Coding Language: Python 3.5
- IDE: Anaconda3
- Application Domain: Machine Learning

3.3 About Python

Python is a high-level, interpreted, interactive and object-oriented scripting language It was created by Guido van Rossum, and released in 1991. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python Features:

1. Easy to learn and use
2. Interrupted language
3. Cross platform Language
4. Free and Open Source
5. Object Oriented Language
6. GUI programming Support
7. Integrated
8. Expressive Language
9. Large Standard Library
10. Extensible

3.4 Natural Language Toolkit

NLTK is one of the leading platforms for working with human language data and Python, the module NLTK is used for natural language processing. NLTK is literally an acronym for Natural Language Toolkit. It provides easy-to-use interfaces and text processing libraries for classification, tokenization, stemming, tagging etc.

NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project. NLTK has been called "a wonderful tool for teaching, and working in, computational linguistics using Python," and "an amazing library to play with natural language." Natural Language Processing with Python provides a practical introduction to programming for language processing.

Lemmatization is the process of grouping together the different inflected forms of a word so they can be analysed as a single item. Lemmatization is similar to stemming but it brings context to the words. So, it links words with similar meaning to one word.

Stemming programs are commonly referred to as stemming algorithms or stemmers. A stemming algorithm reduces the words "chocolates", "chocolatey", "choco" to the root word, "chocolate" and "retrieval", "retrieved", "retrieves" reduce to the stem "retrieve".

3.5 Libraries

Python's standard library is very extensive The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. The Python installers for the Windows platform usually include the entire standard library and often also include many additional components. For Unix-like operating systems Python is normally provided as a collection of packages, so it may be necessary to use the packaging tools provided with the operating system to obtain some or all of the optional components.

Some of the libraries are:

1. **NumPy:** NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. To import NumPy, command used is "import numpy as np.

2. **Pandas:** Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the

three-clause BSD license. In this project pandas will solely be used for importing dataset. To import pandas, command used is “import pandas as pd”.

3. **Matplotlib:** Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using generalpurpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. To import Matplotlib, command used is “import matplotlib.pyplot as plt”.

4. **Seaborn:** Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. To import Seaborn, command used is “import seaborn as sns”.

5. **Scikit-learn:** Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. To import Scikit-learn, command used is “import sklearn”.

3.6 Functional Requirement

In software engineering a functional requirement defines a software system or its component. A function is defined as a set of inputs the behaviour and outputs. Functional requirement may be calculations, technical details, data manipulation and processing and also specify what a system is supposed to be accomplished.

INPUT: Implement pancreatitis patients data set.

PROCESS: pre-process data, implementing algorithms and predicting results.

OUTPUT: predictive representations of recommender system.

3.7 Non-Functional Requirement

Non-Functional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

The major non-functional requirements of the system are as follows.

Usability: - The system is designed with completely automated process to provide the best result to the user hence there is no or less user intervention.

Reliability: -The system is more reliable because of the qualities that are inherited from the Chosen platform. The code built by using Python is more reliable.

Supportability: -The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware.

Data integrity: -Data integrity is the maintenance of, and the assurance of the accuracy and consistency of, data over its entire life-cycle and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data. It is at times used as a proxy term for data quality while data validation is a pre-requisite for data integrity. Data integrity is the opposite of data corruption.

Adaptability: -Adaptability is a feature of a system or of a process. This word has been put to use as a specialized term in different disciplines and in business operations. In ecology, adaptability has been described as the ability to cope with unexpected disturbances in the environment.

Accessibility: -The main goal of the project is Accessibility. We should design our project to set any device and easy to access. Our simple to access with minimum bandwidth internet connection.

4. System Analysis

4.1 Proposed System

Machine learning classification problem in which a voice gender detection dataset. This database was created to identify a voice as male or female, based upon acoustic properties of the voice and speech. The dataset consists of 3,168 recorded voice samples, collected from male and female speakers. Some of the columns present in the dataset are mean frequency, standard deviation, skewness, Kurtosis, spectral entropy, spectral flatness. The data set is basically about voice detection which is recently updated a month ago downloaded from Kaggle.com for a better accurate result to be generated. In this paper we have used Classification Algorithms such as KNN, Logistic Regression, Decision Tree, SVM, Naïve Bayes Algorithms. The data set is divided mainly into trained data and test data. Hence the main focus is on splitting the data into two part that is relying features i.e. X and output feature i.e. Y. The resulting program achieves 98% accuracy on the test set.

We took the Voice gender detection dataset and did the pre-processing after that data cleaning is done. Then used training model and we tested the models. We have applied Classification techniques like SVM, Logistic Regression, decision tree, random forest, Naive bayes and KNN. From this, we have collected the results and finally analysis has done.

The advantage of the system is to reduce the training time, increase accuracy and reduce overfitting when model is created to predict a gender recognition system which is a Machine Learning Classification problem.

4.2 Feasibility Analysis

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities.

The following are its features:

Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input,

output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed. The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project.

Economic Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

The costs conduct a full system investigation, cost of the hardware and software and benefits in the form of reduced costs or fewer costly errors.

Behavioural Feasibility

It checks with the user support, efficiency and harm to the system. The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

5. System Design

5.1 System Architecture

a) Data Pre-Processing

Raw data is highly susceptible to noise, missing values, and inconsistency. The quality of data affects the machine learning results. The raw data is pre-processed to improve the efficiency and ease of the machine learning process. Data pre-processing is one of the most critical steps in the machine learning process which deals with the preparation and transformation of the initial dataset.

- **Data Cleaning**

  Data cleaning routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.

  Missing values

  These are various method using which missing values can be handled:

  - **Ignore the tuple:** This is usually done when the class label is missing (assuming the mining task involves classification or description). This method is not very effective, unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably. Here the tuples which have many missing values and didn't have the output variable function are removed manually.
  - **Fill in the missing value manually:** In general, this approach is time-consuming and may not be feasible given a large data set with many missing values.
  - **Use a global constant to fill in the missing value:** Replace all missing attribute values by the same constant, such as a label like ―NaN". If missing values are replaced by, say, ―NaN", then the mining program may mistakenly think that they form an interesting concept, since they all have a value in common - that of ―NaN". Hence, although this method is simple, it is not recommended. So here the missing values are replaced by NaN but will further be filled by a meaning full value.
  - **Use the attribute mean, median or most frequent to fill in the missing value:** No machine learning algorithm takes a dataset containing missing values. So it is important to replace NaN values with a meaning full value. This can be done using sklearn. pre-processing. Imputer package.

There are three way we can fill the values using the above package by giving strategy attributes the following values:

- If "mean", then replace missing values using the mean along the axis.
- If "median", then replace missing values using the median along the axis.

- If “most frequent”, then replace missing using the most frequent value along the axis.

In the proposed project we will be choosing median of the attribute to fill the missing data.

Before finding the missing values we need to find the number of missing values in the dataset

.

```
# to check null value or missing values
df.isnull().values.any()
```

```
False
```

b) Noisy Data

Noise is a random error or variance in a measured variable. Noisy data can be handled using any of the following methods:

1. Binning methods:

Binning methods smooth a sorted data value by consulting the neighbourhood, or values around it. The sorted values are distributed into several 'buckets', or bins. Because binning methods consult the neighbourhood of values, they perform local smoothing.

2. Clustering:

Outliers may be detected by clustering, where similar values are organized into groups or clusters. Intuitively, values which fall outside of the set of clusters may be considered outliers.

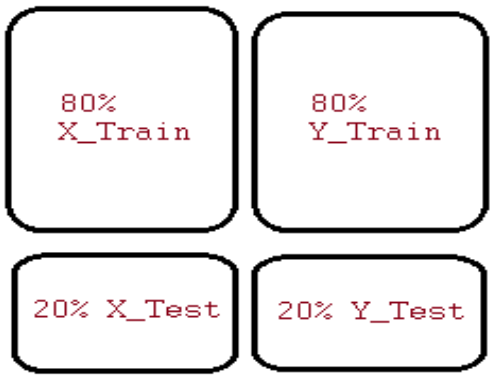
3. Combined computer and human inspection:

Outliers may be identified through a combination of computer and human inspection. In one application, for example, an information-theoretic measure was used to help identify outlier patterns in a handwritten character database for classification. The measure's value reflected the ―surprise" content of the predicted character label with respect to the known label. Outlier patterns may be informative or ―garbage". Patterns whose surprise content is above a threshold are output to a list. A human can then sort through the patterns in the list to identify the actual garbage ones.

c) Preparing the Data for Algorithm

The Data set is divided into two-part X and Y, here X contain all the features except the outcome feature that is hf_score. And Y contain only the outcome feature that is hf_score.
After data is divided vertically. It is divided horizontally into train data and test data i.e. X_train, X_test and Y_train, Y_Test.

80% of the data is divided into train data and 20% as test data. The train data will be used to train the algorithm and test data will be used to test the accuracy of the data.



The Data is divided into X and Y first where X is the independent feature and Y is the dependent feature

d) Feature Selection

Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. The data features that you use to train your machine learning models have a huge influence on the performance you can achieve. Irrelevant or partially relevant features can negatively impact model performance.

Feature selection and Data cleaning should be the first and most important step of your model designing. Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in. Having irrelevant features in your data can decrease the accuracy of the models and make your model learn based on irrelevant features.

Advantages of Feature Engineering:

i. **Reduces Overfitting:** Less redundant data means less opportunity to make decisions based on noise.

ii. **Improves Accuracy:** Less misleading data means modelling accuracy improves.

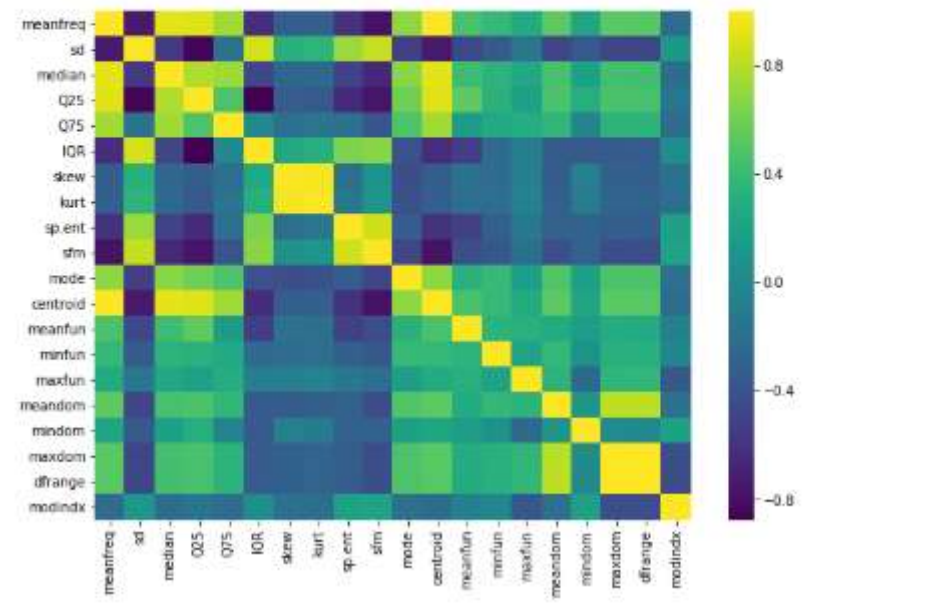
iii. **Reduces Training Time:** fewer data points reduce algorithm complexity and algorithms train faster.

Let's discuss various methodology using which we can reduce the number of features.

1) **Find correlated attributes and using Heatmap:** It is important to determine and compute the degree to which features in dataset are reliant on each other. This understanding can help enhance the data to meet the potential of any machine learning algorithms, such as linear regression, whose performance will reduce with the existence of these interdependencies [4]. The statistical relationship between two features is referred to as correlation. Basically, Correlation are of three types:
    - **Positive Correlation:** Both attributes change in same direction
    - **Neutral Correlation:** No relation in the change of attributes values.
    - **Negative correlation:** Both attributes changes in opposite direction.

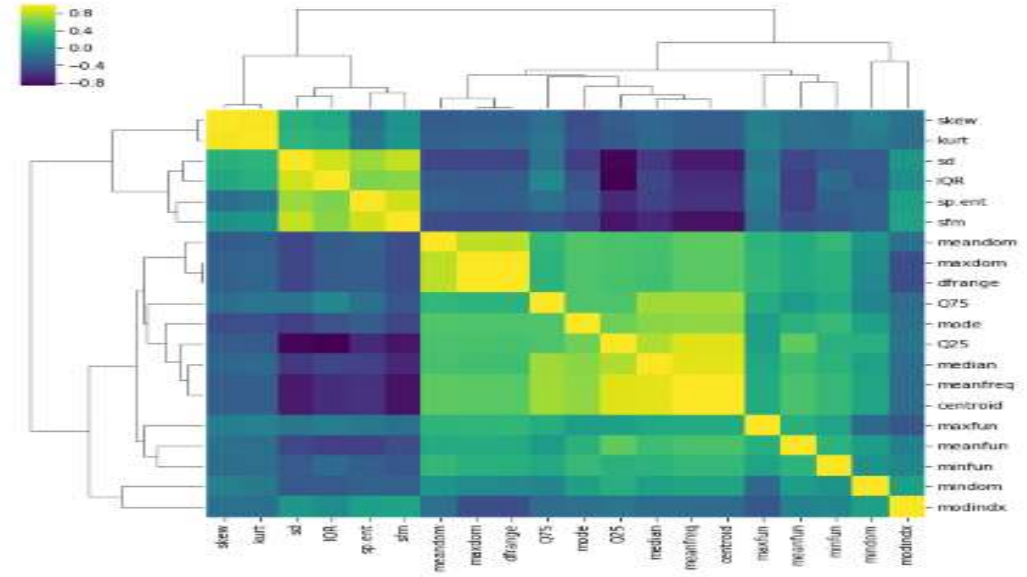
```
#correlation matrix
fig = plt.figure(figsize = (10,7))
ax = fig.gca()
sns.heatmap(df.corr(),ax=ax,cmap="viridis")

#In correlation matrix we can see relation between attributes are negative
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x24d96dd59b0>
```

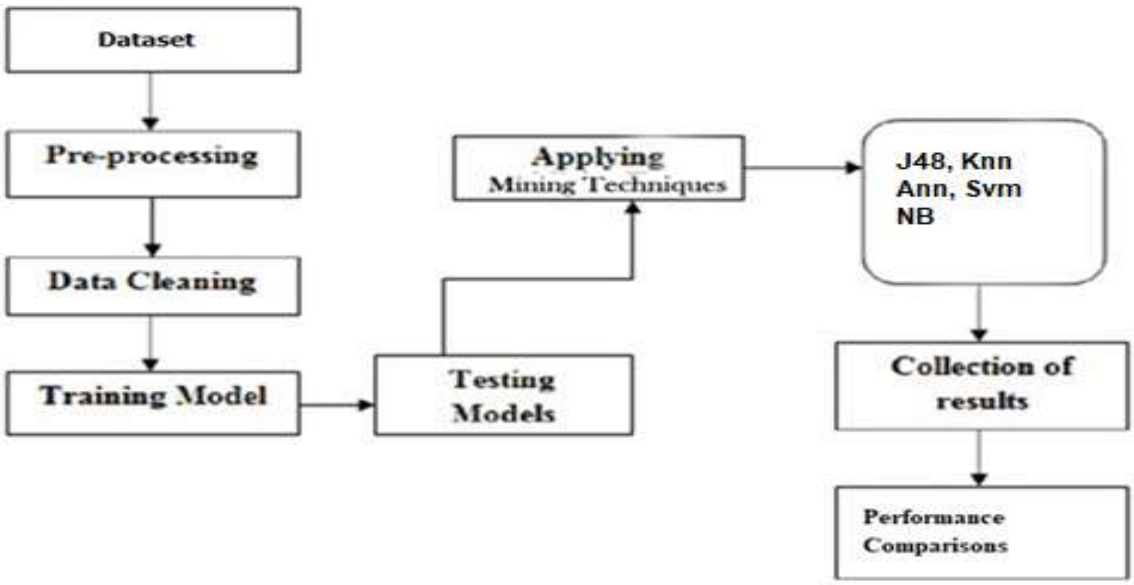


2) **Find correlated attributes and using Cluster Grid plot:**

```
fig = plt.figure(figsize = (10,7))
sns.clustermap(df.corr(),cmap="viridis")
<seaborn.matrix.ClusterGrid at 0x24d98033208>
<Figure size 720x504 with 0 Axes>
```



3) System Architecture for implementation using ML Techniques



5.2 About the Dataset

The Gender Voice Recognition database was created to identify a voice as male or female, based upon acoustic properties of the voice and speech. The dataset consists of 3,168 recorded voice samples, collected from male and female speakers. Some of the columns present in the dataset are mean frequency, standard deviation, skewness, Kurtosis, spectral entropy, spectral flatness.

The data set is basically about voice detection which is recently updated a month ago downloaded from Kaggle.com for a better accurate result to be generated. In this paper we have used Classification Algorithms such as KNN, Logistic Regression, Decision Tree, SVM, Naïve Bayes Algorithms.

The data set is divided mainly into trained data and test data. Hence the main focus is on splitting the data into two part that is relying features i.e. X and output feature i.e. Y. The resulting program achieves 98% accuracy on the test set.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3168 entries, 88 to 1362
Data columns (total 21 columns):
meanfreq    3168 non-null float64
sd          3168 non-null float64
median      3168 non-null float64
Q25         3168 non-null float64
Q75         3168 non-null float64
IQR         3168 non-null float64
skew        3168 non-null float64
kurt        3168 non-null float64
sp.ent      3168 non-null float64
sfm         3168 non-null float64
mode        3168 non-null float64
centroid    3168 non-null float64
meanfun     3168 non-null float64
minfun      3168 non-null float64
maxfun      3168 non-null float64
meandom     3168 non-null float64
mindom      3168 non-null float64
maxdom      3168 non-null float64
dfrange     3168 non-null float64
modindx     3168 non-null float64
label       3168 non-null object
dtypes: float64(20), object(1)
memory usage: 544.5+ KB
```

We took the Voice gender detection dataset and did the pre-processing after that data cleaning is done. Then used training model and we tested the models. We have applied Classification techniques like SVM, Logistic Regression, decision tree, random forest, Naive bayes and KNN. From this, we have collected the results and finally analysis has done.

```
df=pd.read_csv('gender_voice_dataset.csv')
df.describe()
```

	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt	sp.ent	sfm	mode	c
count	3168.000000	3168.000000	3168.000000	3168.000000	3168.000000	3168.000000	3168.000000	3168.000000	3168.000000	3168.000000	3168.000000	3168.
mean	0.180907	0.057126	0.185621	0.140456	0.224765	0.084309	3.140168	36.568461	0.895127	0.408216	0.165282	0.
std	0.029918	0.016652	0.036360	0.048680	0.023639	0.042783	4.240529	134.928661	0.044980	0.177521	0.077203	0.
min	0.039363	0.018363	0.010975	0.000229	0.042946	0.014558	0.141735	2.068455	0.738651	0.036876	0.000000	0.
25%	0.163662	0.041954	0.169593	0.111087	0.208747	0.042560	1.649569	5.669547	0.861811	0.258041	0.118016	0.
50%	0.184838	0.059155	0.190032	0.140286	0.225684	0.094280	2.197101	8.318463	0.901767	0.396335	0.186599	0.
75%	0.199146	0.067020	0.210618	0.175939	0.243660	0.114175	2.931694	13.648905	0.928713	0.533676	0.221104	0.
max	0.251124	0.115273	0.261224	0.247347	0.273469	0.252225	34.725453	1309.612887	0.981997	0.842936	0.280000	0.

```
df.head()
```

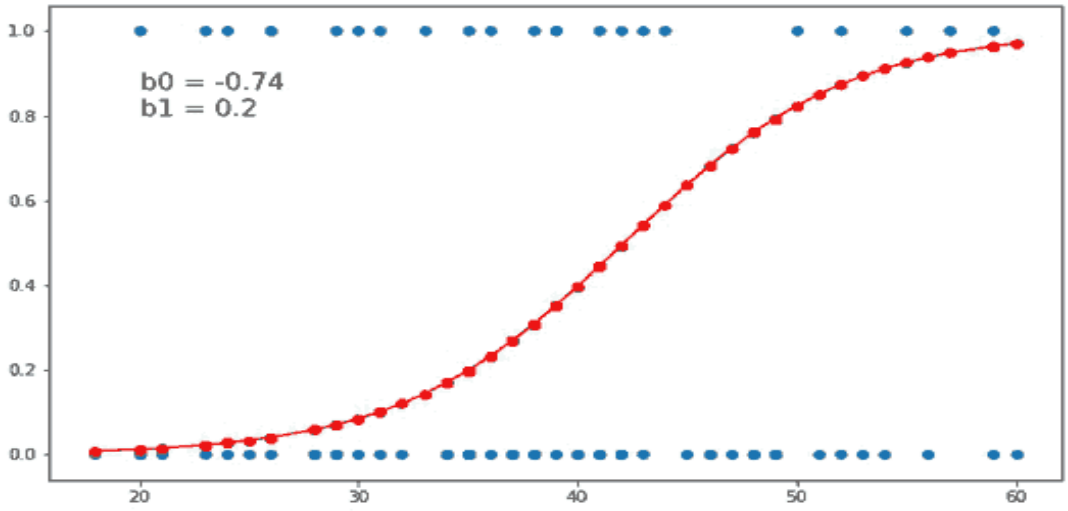
	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt	sp.ent	sfm	...	centroid	meanfun	minfun	maxfun	mean
0	0.059781	0.064241	0.032027	0.015071	0.090193	0.075122	12.863462	274.402906	0.893369	0.491918	...	0.059781	0.084279	0.015702	0.275862	0.00
1	0.066009	0.067310	0.040229	0.019414	0.092666	0.073252	22.423285	634.613855	0.892193	0.513724	...	0.066009	0.107937	0.015826	0.250000	0.00
2	0.077316	0.083829	0.036718	0.008701	0.131908	0.123207	30.757155	1024.927705	0.846389	0.478905	...	0.077316	0.098706	0.015656	0.271186	0.00
3	0.151228	0.072111	0.158011	0.096582	0.207955	0.111374	1.232831	4.177296	0.963322	0.727232	...	0.151228	0.088965	0.017798	0.250000	0.20
4	0.135120	0.079146	0.124656	0.078720	0.206045	0.127325	1.101174	4.333713	0.971955	0.783568	...	0.135120	0.106398	0.016931	0.266667	0.71

5 rows × 21 columns

6. Algorithms

6.1 Logistic Regression

Logistic regression is a machine learning algorithm for classification. In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function. The advantage of logistic regression is, it is designed for this purpose (classification), and is most useful for understanding the influence of several independent variables on a single outcome variable. The disadvantages is it works only when the predicted variable is binary, assumes all predictors are independent of each other, and assumes data is free of missing values.




```
from sklearn.linear_model import LogisticRegression
lr =  LogisticRegression()
lr.fit(x_train, y_train)
y_pred=lr.predict(x_test)
```

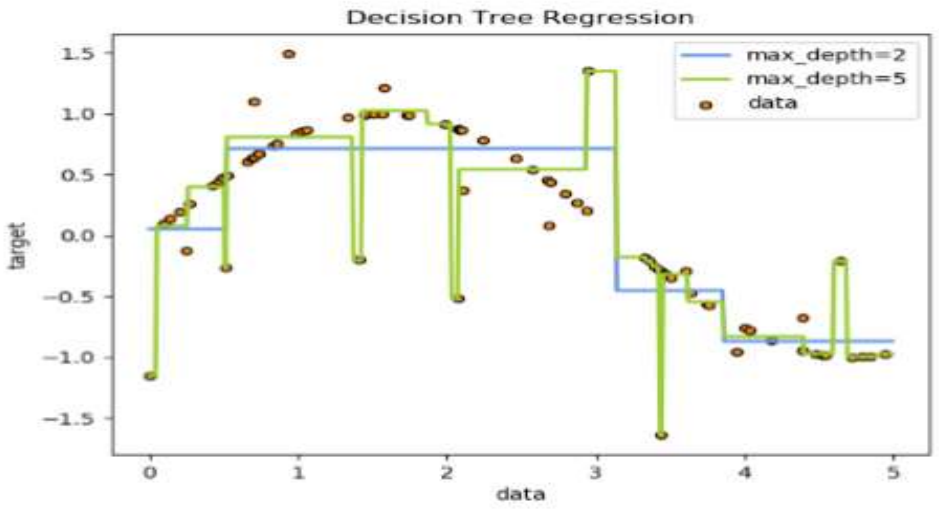
6.2 K-Nearest Neighbours

Neighbours based classification is a type of lazy learning as it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the k nearest neighbours of each point. The advantage is**,** this algorithm is simple to implement, robust to noisy training data, and effective if training data is large. The disadvantage is it Need to determine the value of K and the computation cost is high as it needs to computer the distance of each instance to all the training samples.


```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=15)
knn.fit(x_train,y_train)
y_pred=knn.predict(x_test)
```

6.3 Decision Tree

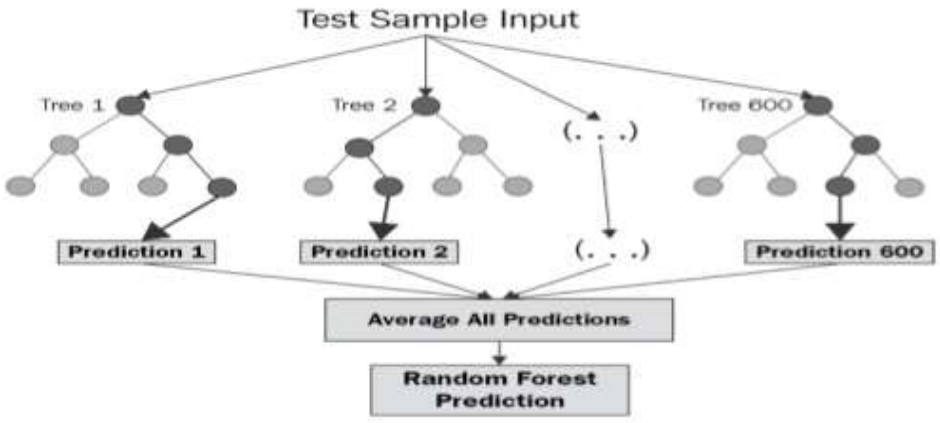
Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data. Decision Tree is simple to understand and visualise, requires little data preparation, and can handle both numerical and categorical data. Decision tree can create complex trees that do not generalise well, and decision trees can be unstable because small variations in the data might result in a completely different tree being generated.



```
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier(max_depth=10, random_state=101,
                               max_features = None, min_samples_leaf = 15)
dtree.fit(x_train, y_train)
y_pred=dtree.predict(x_test)
```

6.4 Random Forest

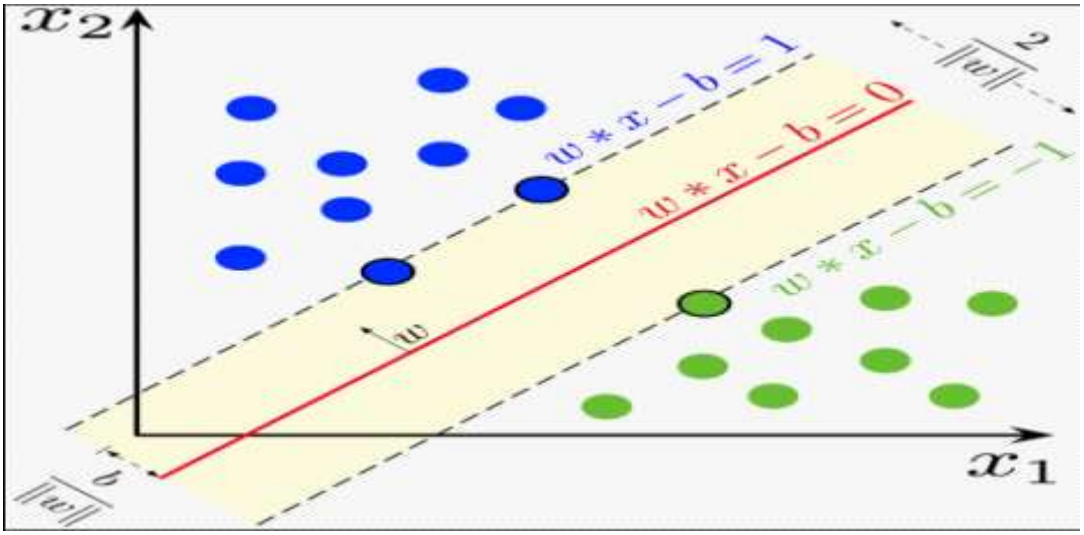
Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement. The advantage is Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases. The disadvantage is Slow real time prediction, difficult to implement, and complex algorithm.




```
from sklearn.ensemble import RandomForestClassifier
rfm = RandomForestClassifier(n_estimators=70, oob_score=True, n_jobs=-1,
                             random_state=101, max_features = None, min_samples_leaf = 30)
rfm.fit(x_train, y_train)
y_pred=rfm.predict(x_test)
```

6.5 Support Vector Machine

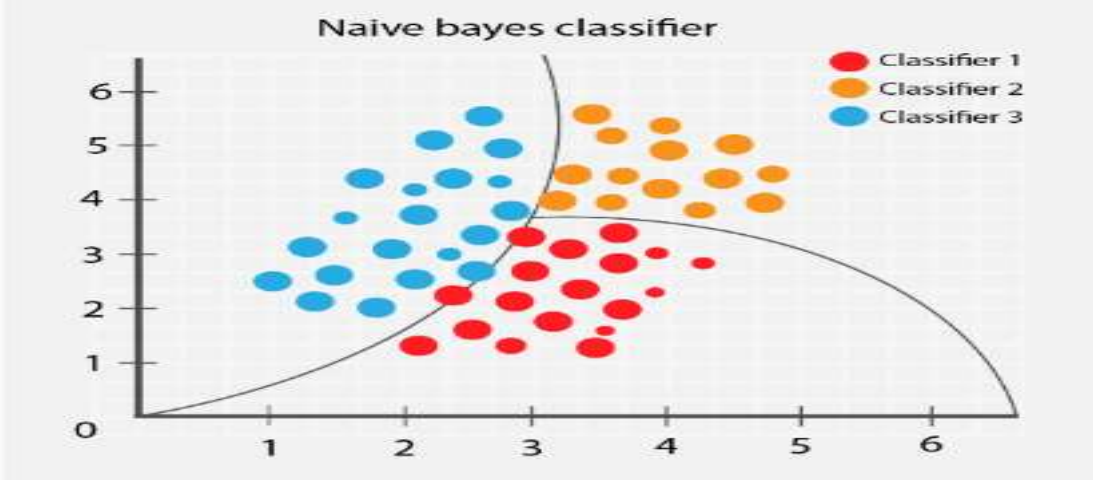
Support vector machine is a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. Support vector machine is effective in high dimensional spaces and uses a subset of training points in the decision function so it is also memory efficient. The algorithm does not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.




```
from sklearn.svm import SVC
svm = SVC(kernel="linear", C=0.025,random_state=101)
svm.fit(x_train, y_train)
y_pred=svm.predict(x_test)
```

6.6 Naives Bayes Algorithm

The Naive Bayes algorithm does that by making an assumption of conditional independence over the training dataset. *The complexity of the above Bayesian classifier needs to be reduced, for it to be practical. This drastically reduces the complexity of above-mentioned problem to just 2n.* The assumption of conditional independence states that, given random variables X, Y and Z, we say X is conditionally independent of Y given Z, if and only if the probability distribution governing X is independent of the value of Y given Z.



7 Sample Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df=pd.read_csv('gender_voice_dataset.csv')
df.describe()

df.head()

df.shape

df.info()

# to check null value or missing values
df.isnull().values.any()

#lets plot count plot for label
sns.countplot(x=df['label'],palette='Set2')

#lets see data distribution
fig = plt.figure(figsize = (15,20))
ax = fig.gca()
df.hist(ax = ax)
plt.tight_layout()
plt.show()

#correlation matrix
fig = plt.figure(figsize = (10,7))
ax = fig.gca()
sns.heatmap(df.corr(),ax=ax,cmap="viridis")

#in correlation matrix we can see relation between attributes are negative

fig = plt.figure(figsize = (10,7))
sns.clustermap(df.corr(),cmap="viridis")
#extract features and labels
df=df.sample(frac=1)
X=df.iloc[:,:20]
y=df['label']

#lets convert the labels into unique integer
from sklearn.preprocessing import LabelEncoder
lbl=LabelEncoder()
```

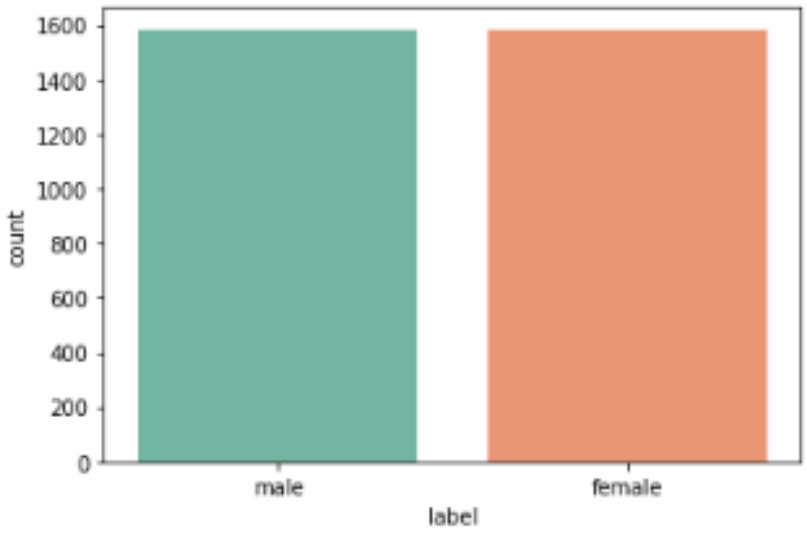
```
y=lbl.fit_transform(y)
y
#male convert to 1
#female convert to 0

#split the dataset into train and test
from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)

X_train.shape,X_test.shape,y_train.shape,y_test.shape

#Logistic Regression

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,roc_auc_score

#accuracy and roc curve
acc_scores=[]
roc_scores=[]

clf=LogisticRegression()
clf.fit(X_train,y_train)
clf.score(X_train,y_train)
y_pred=clf.predict(X_test)
acc_scores.append(accuracy_score(y_test,y_pred))
roc_scores.append(roc_auc_score(y_test,y_pred))
acc_scores[0],roc_scores[0]

#pd.DataFrame(data={'Actual':y_test,'Predict':y_pred}).head


#Support Vector
from sklearn.svm import SVC

clf=SVC(kernel='linear',gamma='scale')
clf.fit(X_train,y_train)
clf.score(X_train,y_train)
y_pred=clf.predict(X_test)
acc_scores.append(accuracy_score(y_test,y_pred))
roc_scores.append(roc_auc_score(y_test,y_pred))
acc_scores[1],roc_scores[1]
#pd.DataFrame(data={'Actual':y_test,'Predict':y_pred}).head

#Naivye Bayes
from sklearn.naive_bayes import GaussianNB
```

```
clf=GaussianNB()
clf.fit(X_train,y_train)
clf.score(X_train,y_train)
y_pred=clf.predict(X_test)
acc_scores.append(accuracy_score(y_test,y_pred))
roc_scores.append(roc_auc_score(y_test,y_pred))
acc_scores[2],roc_scores[2]
#pd.DataFrame(data={'Actual':y_test,'Predict':y_pred}).head

#Decision Tree
from sklearn.tree import DecisionTreeClassifier

clf=DecisionTreeClassifier()
clf.fit(X_train,y_train)
clf.score(X_train,y_train)
y_pred=clf.predict(X_test)
acc_scores.append(accuracy_score(y_test,y_pred))
roc_scores.append(roc_auc_score(y_test,y_pred))
acc_scores[3],roc_scores[3]
#pd.DataFrame(data={'Actual':y_test,'Predict':y_pred}).head

#KNN
from sklearn.neighbors import KNeighborsClassifier

clf=KNeighborsClassifier(n_neighbors=33,algorithm='ball_tree')
clf.fit(X_train,y_train)
clf.score(X_train,y_train)
y_pred=clf.predict(X_test)
acc_scores.append(accuracy_score(y_test,y_pred))
roc_scores.append(roc_auc_score(y_test,y_pred))
acc_scores[4],roc_scores[4]
#pd.DataFrame(data={'Actual':y_test,'Predict':y_pred}).head

#Random Forest
from sklearn.ensemble import RandomForestClassifier

clf=RandomForestClassifier(n_estimators=150)
clf.fit(X_train,y_train)
clf.score(X_train,y_train)
y_pred=clf.predict(X_test)
acc_scores.append(accuracy_score(y_test,y_pred))
roc_scores.append(roc_auc_score(y_test,y_pred))
acc_scores[5],roc_scores[5]
#pd.DataFrame(data={'Actual':y_test,'Predict':y_pred}).head
```

```
#Random forest has the highgest accuracy

fig = plt.figure(figsize = (9,7))
ax = fig.gca()
plt.bar(['Logistic','SVC','Naivye Bayes','Decision Trees','KNN','Random Forest'],acc_scores,width=0.7,color=('Red','Maroon','Purple','SkyBlue','Orange','Brown'))
plt.xlabel('Algortihms')
plt.ylabel('Accuracy')
plt.show()


fig = plt.figure(figsize = (9,7))
ax = fig.gca()
plt.bar(['Logistic Regression','SVC','Naivye Bayes','Decision Trees','KNN','Random Forest'],roc_scores,width=0.7,color=('Red','Maroon','Purple','SkyBlue','Orange','Brown'))
plt.xlabel('Algorithms')
plt.ylabel('ROC AUC Scores')
plt.show()
```

8 Screenshot

```
<matplotlib.axes._subplots.AxesSubplot at 0x20ae41cb080>
```

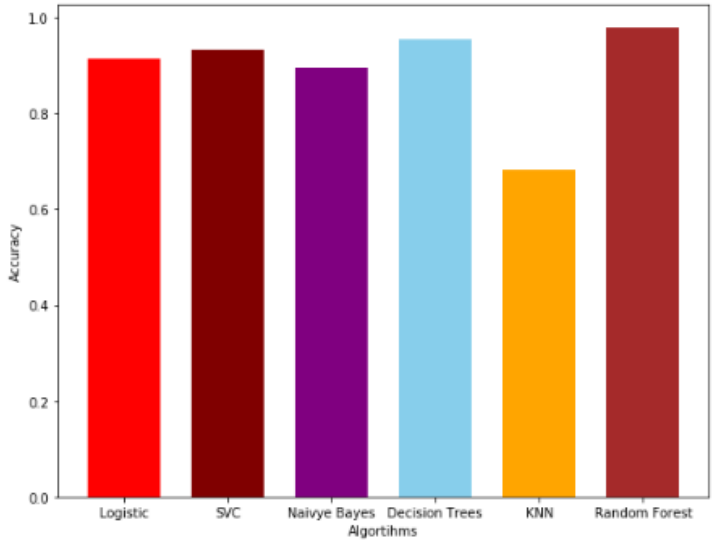


Screenshot: Count of Gender

```
#Random Forest
from sklearn.ensemble import RandomForestClassifier

clf=RandomForestClassifier(n_estimators=150)
clf.fit(X_train,y_train)
clf.score(X_train,y_train)
y_pred=clf.predict(X_test)
acc_scores.append(accuracy_score(y_test,y_pred))
roc_scores.append(roc_auc_score(y_test,y_pred))
acc_scores[5],roc_scores[5]
#pd.DataFrame(data={'Actual':y_test,'Predict':y_pred}).head


#Random forest has the highgest accuracy
```

```
(0.9779179810725552, 0.9774975345904432)
```

Screenshot: The best Accuracy Prediction (98%)



Screenshot: Bar Graph

```
In [14]: model = pickle.load(open('model.pkl','rb'))
    ...:
print(model.predict([[0.077315503,0.083829421,0.036718459,0.008701057,0.131908017,0.12320696
1,30.75715458,1024.927705,0.846389092,0.478904979,0,0.077315503,0.098706262,0.015655577,0.27
1186441,0.007990057,0.0078125,0.015625,0.0078125,0.046511628]]))
[1]

In [15]: model = pickle.load(open('model.pkl','rb'))
    ...:
print(model.predict([[0.163938529,0.079085581,0.191072961,0.116967096,0.216309013,0.09934191
7,1.971270984,7.669632588,0.931116286,0.587657557,0.207095851,0.163938529,0.168650305,0.0200
50125,0.242424242,0.591619318,0.0078125,3.09375,3.0859375,0.277468354]]))
[0]
```

Screenshot: Prediction as Male [1] or Female [0]

```
In [31]: from sklearn import tree
cls = tree.DecisionTreeClassifier()
x = [[0.077315503,0.083829421,0.036718459,0.008701057,0.131908017,0.123206961,30.75715458,1024.927705,0.846389092,0.478904979,0,
],
    [0.163938529,0.079085581,0.191072961,0.116967096,0.216309013,0.099341917,1.971270984,7.669632588,0.931116286,0.587657557,0.
]]
y = ["Male","Female"]
cls = cls.fit(x,y)
predication = cls.predict([[0.077315503,0.083829421,0.036718459,0.008701057,0.131908017,0.123206961,30.75715458,1024.927705,0.84
]])
print(predication)

['Male']
```

9 Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- it is sufficiently usable,
- can be installed and run in its intended environments, and
- achieves the general result its stakeholders desire.

TYPES OF TESTING

1. UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive.

Test objectives:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

2. INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one steps up – software applications at the company level – interact without error.

Test Results: All the test cases should be passed successfully. No defects encountered.

3. ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

4. FUNCTIONAL TESTING

Functional Testing is a type of software testing whereby the system is tested against the functional requirements/specifications. Functional testing ensures that the requirements are properly satisfied by the application. ensures that the requirements are properly satisfied by the application.

5. SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6. WHITEBOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7. BLACKBOX TETSING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software

under test is treated, as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Mentioned below are critical activities that are essential to test machine learning systems:

1. Developing training data sets: This refers to a data set of examples used for training the model. In this data set, you have the input data with the expected output. This data is usually prepared by collecting data in a semi-automated way.

2. Developing test data sets: This is a subset of the training dataset that is intelligently built to test all the possible combinations and estimates how well your model is trained. The model will be fine-tuned based on the results of the test data set.

3. Developing validation test suites based on algorithms and test datasets. Taking the DNA example, test scenarios include categorizing patient outcomes based on DNA sequences and creating patient risk profiles based on demographics and behaviours.

4. The key to building validation suites is to understand the algorithm. This is based on calculations that create a model from the training data. To create a model, the algorithm analyses the data provided, looks for specific patterns, and uses the results of this analysis to develop optimal parameters for creating the model. The model is refined as the number of iterations and the richness of the data increase. Some algorithms in popular use are regression algorithms that predict one or more continuous numeric variables such as return on investment. Another example is the association of algorithms that create co-relations based on attributes of a data set. This is used for portfolio analysis in capital markets. Another illustration in digital applications is sequence algorithms that predict customer behaviour based on a series of clicks or paths on a digital platform.

5. Communicating test results in statistical terms. Testers are traditionally used to expressing the results of testing in terms of quality such as defect leakage or severity of 48 defects. Validation of models based on machine algorithms will produce approximations and not exact results. The testing community will need to determine the level of confidence within a certain range for each outcome and articulate the same. So, in this project the entire dataset is divided into training data and testing data. The test data is given to the model to test the accuracy which the data provide. The training dataset contain 80% of the entire data in the dataset and testing data contain 20% of the entire data in the dataset.

When the model is created and test data is given to the model for prediction, we get the following accuracy of different regression model:

Algorithm	Train Data	Test Data
Random Forest Algorithm	0.97	0.97
Decision Tree Algorithm	0.96	0.96
Logistic Regression Algorithm	0.90	0.90
Support Vector Machine	0.91	0.91
Naïve Bayes Algorithm	0.86	0.86
KNN Algorithm	0.69	0.69

Hence it is clear the Random Forest model give the maximum accuracy of 98% using the test data and hence is the best model for the given problem statement.

10 Conclusion

By studying several algorithms on the given dataset Random Forest Algorithm gives the accurate and perfect results when comparing with the other Classification Algorithms.

Finally, to identify a voice as male or female based upon acoustic properties of the voice and speech.

Algorithm	Train Data	Test Data
Random Forest Algorithm	0.97	0.97
Decision Tree Algorithm	0.96	0.96
Logistic Regression Algorithm	0.90	0.90
Support Vector Machine	0.91	0.91
Naïve Bayes Algorithm	0.86	0.86
KNN Algorithm	0.69	0.69

It is concluded that, when feature engineering is done on gender voice recognition dataset, it helps to reduce the number of features to only features which are of high importance. Once the feature engineering is done it helps the machine learning classification algorithms to train the data in a lesser time than before. When model was trained before feature engineering it took more time than after feature engineering was done.

For future scope, implementation of a voice-based emotion detection system it presents an opportunity to implement it over smart phone platforms. And it can be used in smart home, smart office and virtual reality, and it may acquire importance in all aspects of future people's life.

11 Bibliography

1) H. Harb and L. Chen, Voice-based gender identification in multimedia applications, Journal of Intelligent Information Systems, 24(2), 179-198 (2005).
2) Md. Sadek Ali1, Md. Shariful Isla1 and Md. AlamgirHossain, Gender recognition of speech signal, International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), 2(1), 1–9 (2012).
3) D. Ververidis and C. Kotropoulos. Automatic speech classification to five emotional states based on gender information. In Proc. XII European Signal Processing Conf., volume 1, pages 341–344. Vienna, Austria,(2004).
4) M. Sedaaghi, A Comparative Study of Gender and Age Classification in Speech Signals, Iranian Journal of Electrical & Electronic Engineering, 5(1), 1–12 (2009)
5) S. Gaikwad, B. Gawali, and S.C. Mehrotra, Gender identification using SVM with combination of MFCC, Advances in Computational Research, 4(1), 69-73, 2012.
6) Y.-M. Zeng, Z.-Y. Wu Falk, and W.-Y. Chan, "Robust gmm based gender classification using pitch and rasta-plp parameters of speech," in Machine Learning and Cybernetics, 2006 International Conference on. IEEE 2006
7) C.S. Leung, M. Lee, and J.H. Chan (Eds.), "Gender Identification from Thai Speech Signal Using a Neural Network" ICONIP 2009, Part I, LNCS 5863, pp. 676–684, 2009
8) Kumar R.,Dutta S.,Kumara shama,"Gender Recognition using speech processing technique using LABVIEW" IJAET May 2011
9) Md. Rabiul Islam1, Md. Fayzur Rahman,"Improvement of Text Dependent Speaker Identification System Using Neuro-Genetic Hybrid Algorithm in Office Environmental Conditions ",IJCSI International Journal of Computer Science Issues, Vol. 1, 2009.
10) Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet," Front-End Factor Analysis for Speaker Verification" IEEE transaction on Audio, Speech, And Language Processing, Vol. 19, No. 4, May 2011.
11) Tomi Kinnunen, " Spectral Features for Automatic TextIndependent Speaker Recognition", Ph. Lic. Thesis, Department of Computer Science University of Joensuu , 2004.
12) Milan Sigmund. "Gender Distinction Using Short Segments Of Speech Signal".
13) Atal B., "Effectiveness of linear prediction characteristics of speech wave for automatic speaker identification and verification", Journal of the acoustic Society of America 1974, pp. 55(6):1304-1312.
14) H. Harb and L. Chen, "Voice-based gender identification in multimedia applications," Journal of Intelligent Information Systems, vol. 24,no. 2, pp. 179–198, 2005.
15) S. McCandless, "An algorithm for automatic formant extraction using linear prediction spectra," IEEE Transactions on acoustics, speech and signal processing, vol. 22, no. 2, pp. 135–141, 1974.
16) J. M. Naik, L. P. Netsch, and G. R. Doddington, "Speaker verification over long distance telephone lines", IEEE Proceedings of the 1989 International Conference on Acoustics, Speech and Signal Processing, Glasgow, Scotland, May 1989, pages 524--527

Annexure-I

Published Journal Paper

International Journal of Research in Advent Technology, Vol.8, No.4, April 2020
E-ISSN: 2321-9637
Available online at www.ijrat.org

Gender Voice Recognition with Classification approach using Random Forest and Decision Tree Algorithms

Mukesh Tadi, S Prasad Babu Vagolu, and Sunil Chandolu

Abstract— Gender identification is one of the major problems of the speech processing. Gender tracking from aural data like median, frequency, and pitch. Machine learning provides auspicious results for the problem of classification in all domains. There are a few standards to work on to appraise the algorithms. Our model comparisons algorithm for appraising different learning algorithms is based on different metrics for classifying gender and aural data. An important parameter in evaluating any algorithms is their performance. The degree of variability should be low for classification set of problems; means the accuracy rate should be pretty high. The position and gender of the person became pretty important in financial markets by the form of AdSense. With this model comparisons algorithm, we tried different ML algorithms and came up with the best fit for the gender classification of aural data.

Index Terms—Gender identification, Voice Recognition, Random Forest Algorithm, Decision Tree Algorithm**.**

I. INTRODUCTION

Finding someone's gender based on their voice is an easy task. In the real world, the difference between male and female voices can easily be identified by human ear in first couple of words. Its most common communication in the world. The voice is full of many linguistic features. These voice features are considered a voice print to recognize [14] the speaker's sex. Voice recordings are considered as input to the system, which is then the system's process for detecting voice features [1]. However, programming to do this becomes very difficult.

Manuscript revised on April 30, 2020 and published on May 10, 2020
***Mukesh Tadi**, PG Student, Department of Computer Science, GITAM (Deemed to be University), Visakhapatnam, India. Email: mukesh.tadi@gmail.com.*
***S.Prasad Babu Vagolu**, Assistant Professor, Department of Computer Science, GITAM (Deemed to be University), Visakhapatnam, India. Email: prasadray@live.com*
***Sunil Chandolu**, Assistant Professor, Department of Computer Science, GITAM (Deemed to be University), Visakhapatnam, India. Email: kunny0306@gmail.com*

This document describes the design of a computer program to illustrate the analysis of words and words that determine gender [2]. Test the input and compare it with the trained model, perform the calculations according to the algorithm used and give the same result i.e. male or female.

II. CLASSIFICATION ALGORITHMS DESCRIPTION

A. Random Forest Algorithm

Random Forest is a supervised learning method used for classification and regression. It is mainly used to collect news of unsafe separation. Each tree provides the cohesion of that feature. The forest chooses a section with the most votes in a particular ward. It is a spherical study of the classification [3] [13], registration and other functions, which works by constructing multiple decision trees during training and extracting a game path (categorization) or mean prediction (repositioning) of individual trees.

B. Decision Tree Algorithm

Decision Tree is also a supervised machine learning technique for both the predictions as well as the classification in machine learning. Tree decisions are trees that are categorized according to feature values. Each location in the decision tree represents a specific element in the image, and each branch represents a value that can be considered a negative space. The tree learning curve, used in data mining and machine learning, uses the decision tree as a model for mapping an object to a specific object to draw conclusions about the value of an object.

C. Logistic Regression Algorithm

Logistic Regression is also for classification problems; is a prediction-based algorithm for analysis and is based on the assumption of probability. Logistic Regression uses a very expensive function, this cost function can be defined as a 'Sigmoid function' or also known as a 'function logistic' instead of a linear function. Other examples of problems with spam emails or not online spam Scanning or Not Fraud, Tumor Malignant or Benign.

D. Support Vector Machine

SVM is also a good for both classification and regression challenges. However, SVM usage is widely in separation problems. In this algorithm, we plot each data element as a point in the n-dimensional space (where the value of n is the

doi: 10.32622/ijrat.84202009

number of elements) for the value of each element that is the sum of a particular combination.

E. Naïve Bayes Algorithm

The Naive Bayes algorithm works by making assumptions of conditional independence in the training data. The complexity of the Bayesian algorithm above needs to be minimized, in order to work. This greatly reduces the severity of the aforementioned problem to only 2n. The assumption of conditional freedom means that, given a random distribution of X, Y and Z, we say that X is an independent of the terms Y by given Z, and only if the probability distribution of the control X is independent of the value of Y by given Z.

F. KNN Algorithm

K-nearest neighbor algorithm can also be useful for classification and prediction problems. Algorithms around K-used (KNN) use 'similarity factor' to estimate the positions of new data points which means that the new data point will be assumed and assigned a value based on nearest training set points. However, main usage is for classification problems in the industry.

III. METHODOLOGY

Here is the problem of machine partitioning when there is data to test for voice sex. This data was created to identify the voice as gent or lady, based on voice and speech properties [10]. Our database consists of 3,168 recorded voice models, collected from gent and lady speakers [16]. The other columns in the dataset are mentioned frequency, standard deviation, deception, Kurtosis, spectral penetration, visual acuity [11]. Supported data is about a voice recovery that was updated later last month from Kaggle.com to get a better accurate result. In this paper we have used Classified Algorithms such as KNN, Logistic Regression [12], Tree Decision, SVM[9], Naïve Bayes Algorithms [15]. The data set is mainly categorized as trained data and test data. Thus, the main focus in dividing the data into two parts is dependent on the signals i.e. X and output factor i.e. Y. An efficient system achieves 89% accuracy in the test set.

We took the Voice gender detection dataset and did the pre-processing after that data cleaning is done [4] [8]. Then used training model and we tested the models. We have applied Classification techniques like Logistic Regression, SVM, random forest, decision tree, KNN and Naive bayes. From this, we have collected the results and finally analysis has done [5] [6] .

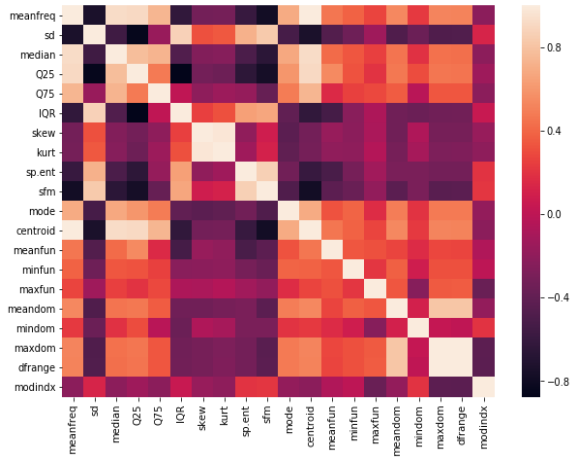


Figure 1 - Correlated Attributes using Heatmap

IV. RESULTS

The test results will be shown below:

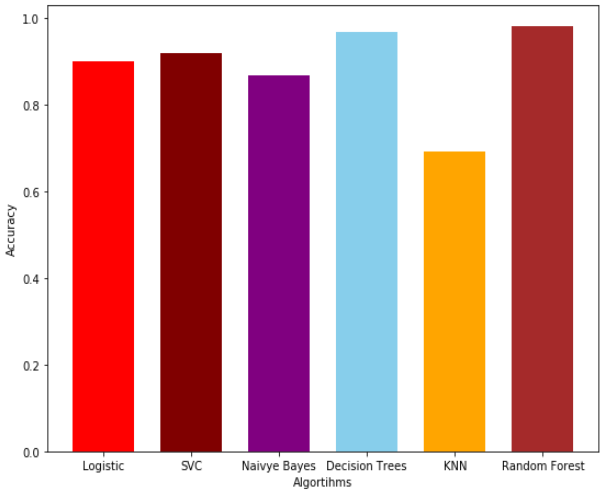


Figure 2 – Bar Graph

Table- I: Performance Comparison Chart of Classification Algorithms-

Algorithm Used	Train data	Test data
Random Forest Algorithm	0.9794	0.9794
Decision Tree Algorithm	0.9668	0.9668
Logistic Regression Algorithm	0.9006	0.9024
Support Vector Machine	0.9179	0.9195
Naive Bayes Algorithm	0.8675	0.8679
KNN Algorithm	0.6924	0.6937

doi: 10.32622/ijrat.84202009

V. CONCLUSION AND FUTURE SCOPE

By studying several algorithms on the given dataset as input in Table- I. Random Forest Algorithm gives the accurate and perfect results when comparing with the other Classification Algorithms. Finally, to identify a voice as male or female based upon acoustic properties of the voice and speech.

In future, the implementation can be done with a voice-based emotion detection system on smart phone platforms. And it can be applied in homes to make it as a smart home, offices as smart office and virtual reality. It may grab the crucial part in all aspects of people's life in the future.

REFERENCES

[1] H. Harb and L. Chen, Voice-based gender identification in multimedia applications, Journal of Intelligent Information Systems, 24(2), 179-198 (2005).
[2] Md. Sadek Ali1, Md. Shariful Isla1 and Md. AlamgirHossain, Gender recognition of speech signal, International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), 2(1), 1–9 (2012).
[3] D. Ververidis and C. Kotropoulos. Automatic speech classification to five emotional states based on gender information. In Proc. XII European Signal Processing Conf., volume 1, pages 341–344. Vienna, Austria,(2004).
[4] M. Sedaaghi, A Comparative Study of Gender and Age Classification in Speech Signals, Iranian Journal of Electrical & Electronic Engineering, 5(1), 1–12 (2009)
[5] S. Gaikwad, B. Gawali, and S.C. Mehrotra, Gender identification using SVM with combination of MFCC, Advances in Computational Research, 4(1), 69-73, 2012.
[6] Y.-M. Zeng, Z.-Y. Wu Falk, and W.-Y. Chan, "Robust gmm based gender classification using pitch and rasta-plp parameters of speech," in Machine Learning and Cybernetics, 2006 International Conference on. IEEE 2006
[7] C.S. Leung, M. Lee, and J.H. Chan (Eds.), "Gender Identification from Thai Speech Signal Using a Neural Network" ICONIP 2009, Part I, LNCS 5863, pp. 676–684, 2009
[8] Kumar R.,Dutta S.,Kumara shama,"Gender Recognition using speech processing technique using LABVIEW" IJAET May 2011
[9] Md. Rabiul Islam1, Md. Fayzur Rahman,"Improvement of Text Dependent Speaker Identification System Using Neuro-Genetic Hybrid Algorithm in Office Environmental Conditions ",IJCSI International Journal of Computer Science Issues, Vol. 1, 2009.
[10] Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet," Front-End Factor Analysis for Speaker Verification" IEEE transaction on Audio, Speech, And Language Processing, Vol. 19, No. 4, May 2011.
[11] Tomi Kinnunen, " Spectral Features for Automatic TextIndependent Speaker Recognition", Ph. Lic. Thesis, Department of Computer Science University of Joensuu , 2004.
[12] Milan Sigmund. "Gender Distinction Using Short Segments Of Speech Signal".
[13] Atal B., "Effectiveness of linear prediction characteristics of speech wave for automatic speaker identification and verification", Journal of the acoustic Society of America 1974, pp. 55(6):1304-1312.
[14] H. Harb and L. Chen, "Voice-based gender identification in multimedia applications," Journal of Intelligent Information Systems, vol. 24,no. 2, pp. 179–198, 2005.
[15] S. McCandless, "An algorithm for automatic formant extraction using linear prediction spectra," IEEE Transactions on acoustics, speech and signal processing, vol. 22, no. 2, pp. 135–141, 1974.
[16] J. M. Naik, L. P. Netsch, and G. R. Doddington, "Speaker verification over long distance telephone lines", IEEE Proceedings of the 1989 International Conference on Acoustics, Speech and Signal Processing, Glasgow, Scotland, May 1989, pages 524--527.

AUTHORS PROFILE

Mukesh Tadi Pursuing Master of Computer applications, Dept of Computer Science, GIS, GITAM(Deemed to be University),Visakhapatnam, Andhra Pradesh, India. His area of interest in python, Data Mining and Machine Learning.

S Prasad Babu Vagolu is an Assistant Professor in Department of Computer Science, GIS, GITAM. He has 8 years of Software Development Experience and 10 years of Teaching Experience. He is CSI lifetime member and passionate about research in Wireless Mesh Networks, Machine Learning and Artificial Intelligence.

Snuil Chandolu is an Assistant Professor in Department of Computer Science, GIS, GITAM. He has 8 years of Teaching Experience. He is passionate about research in Wireless Networks, Machine Learning and Artificial Intelligence.

doi: 10.32622/ijrat.84202009