***GENDER RECOGNITION USING VOICE***

Project report submitted in partial fulfilment of

the Requirements for the

Award of the Degree of B. Tech in

Computer Science and Engineering

BY

**Sparsh Saxena (2013662)**

**Shivani Deoli (2013690)**

Under the guidance of

**Mr. Karan Mehta (Mentor)**

Logo

Description automatically generated

**Department of Computer Science and Engineering**

**Graphic Era Deemed to be University**

**Dehradun-248002**

**2021**



This is to certify that the project report entitled **“*Gender Recognition using Voice*”** being submitted by

1. **Sparsh Saxena (2013662)**
2. **Shivani Deoli (2013690)**

in partial fulfilment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Graphic Era Deemed to be University is a record of bona fide work carried out under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.



|  |  |  |
| --- | --- | --- |
| **INDEX** | **CONTENT** | **PAGE** |
| 1. | Introduction | 4 |
| 2. | Aims | 5 |
| 3. | Objectives | 5 |
| 4. | Problem Summary | 6 |
| 5. | IBM SPSS Modeler | 6 |
| 6. | Phase-I Achievements | 7 |
| 7. | Conclusion | 17 |
| 8. | Contributions | 18 |
| 9. | References | 18 |
|  |  |  |

***Introduction:***

Determining a person’s gender as male or female, based upon a sample of their voice seems to initially be an easy task. Often, the human ear can easily detect the difference between a male or female voice within the first few spoken words. However, designing a computer program to do this turns out to be a bit trickier.

The model is constructed using 3,168 recorded samples of male and female voices, speech, and utterances. The samples are processed using acoustic analysis and then applied to an artificial intelligence/machine learning algorithm to learn gender-specific traits. The resulting program achieves 97% accuracy on the test set.

Table

Description automatically generated

***Aims:***

1. **Back-end:** Create and deploy/export machine learning model trained on provided dataset of acoustic measures of voice (males/females) to use in front-end.
2. **Front-end:** Create a simple UI (User Interface) through which user will interact with our application and use the different functionalities present in our application.

***Objectives:***

1. **Application UI:**
2. User interface should be simple and easy to use.
3. Less number of clicks and keystrokes should be required to accomplish the task.
4. Interface design should be intuitive. Intuitive user interface design is on that is easy to learn so that user can pick it up quickly and easily.
5. **Machine Learning Model:**
6. The model should be light weight.
7. The model should be able to process voice and give results in short amount of time.

***Problem Summary:***

The main aim of the project is to recognize gender of the person by using his/her voice. For this purpose, we will be working on both back-end and front-end for ease of the user. In back-end we will be training a machine learning model using various machine learning algorithms on historical data then making an executable application or a web application for the front-end.

**Technology Used:**

1. **jupyter notebook:** Used jupyter notebook for building back-end from scratch. Used jupyter notebook for training machine learning model using various machine learning algorithms.
2. **IBM SPSS Modeler:** Used IBM SPSS Modeler for analyzing different machine learning model and selecting the best algorithms for training our machine learning model in jupyter notebook.

***IBM SPSS Modeler:***

SPSS Modeler is a leading visual data science and machine learning (ML) solution designed to help enterprises accelerate time to value by speeding up operational tasks for data scientists. Organizations worldwide use it for data preparation and discovery, predictive analytics, model management and deployment, and ML to monetize data assets.

IBM SPSS Modeler is a data mining and text analytics software application from IBM. It is used to build predictive models and conduct other analytic tasks. It has a visual interface which allows users to leverage statistical and data mining algorithms without programming

Graphical user interface

Description automatically generated

***Phase-I Achievements:***

In phase-I, we will be only working on back-end of our project that is, training our machine learning model using different technologies and different machine learning algorithms.

To analyze gender by voice and speech, a training database was required. A database was built using thousands of samples of male and female voices, each labeled by their gender of male or female.

Each voice sample is stored as a .WAV file, which is then pre-processed for acoustic analysis using the specan function from the **WarbleR** R package. Specan measures 22 acoustic parameters on acoustic signals for which the start and end times are provided.

The output from the pre-processed WAV files were saved into a CSV file, containing 3168 rows and 21 columns (20 columns for each feature and one label column for the classification of male or female).

Some of the Acoustic properties measured were-:



The data we collected to train our machine learning model was preprocessed and cleaned that is, there were no null values, invalid values, outlier/noise.

To select the appropriate machine learning algorithm, we used IBM SPSS Modeler.

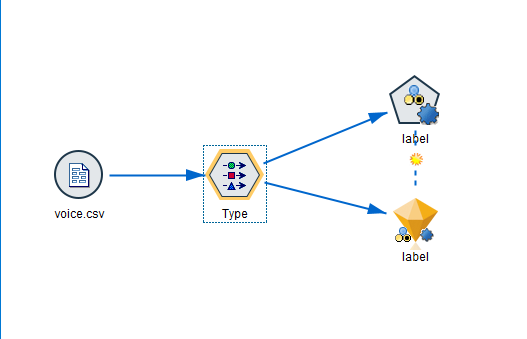
Steps involved to select machine learning algorithm in SPSS Modeler -:

1. Drag and drop source file (voice.csv) in stream canvas in SPSS Modeler.
2. Drag and drop ‘type node’ from ‘Field Ops’ palette to select data type of features available in dataset. And set ‘label’ feature role as target and measurement as Flag.

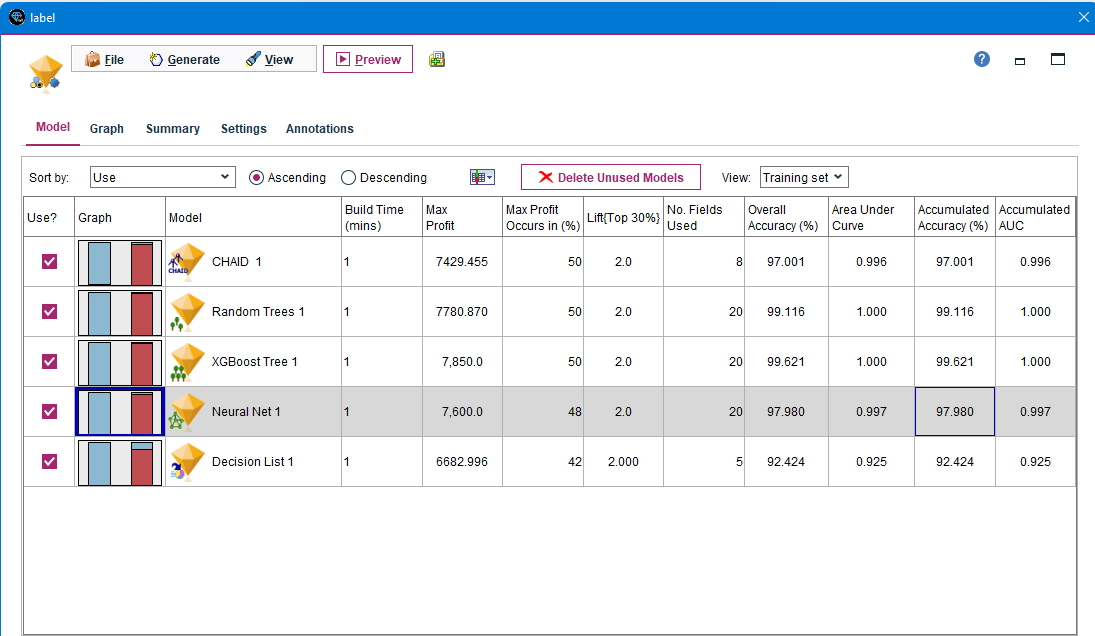
Table

Description automatically generated

1. Link source node and type node for data flow from one node to another node.
2. Drag and drop ‘Auto Classifier’ node in stream canvas from ‘Modeling’ palette to train and select appropriate machine learning algorithm.
3. Link ‘type’ node and ‘Auto Classifier’ Node to complete our stream.
4. Click on run button from the menu. The stream will look like as shown below.



1. After the stream is executed, machine learning models with most accuracy are as shown:



After selecting the appropriate machine learning model, we trained and exported our machine learning model using jupyter notebook. Steps involved in training our machine learning using jupyter notebook were:

1. **IMPORTING IMPORTANT LIBRARIES:**

**Graphical user interface, text, application

Description automatically generated**

1. **LOADING DATASET IN JUPYTER NOTEBOOK:**

**Graphical user interface, text, chat or text message, website

Description automatically generated**

1. **EXPLORING DATA:**

**Graphical user interface

Description automatically generated with medium confidence**

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Here we can see we don’t have any null values, invalid values, and any outlier. After exploring the dataset, we can see that our independent variables/features (20 columns) are continuous and our dependent variable/feature or known as target variable is binary (male or female). So we can convert it into 0 or 1, where 1 signifies Male and 0 signifies Female

1. **ENCODING LABELS:**

To achieve the above, we need to perform encoding and convert our binary classification in string to 0 or 1.

Graphical user interface, application

Description automatically generated

Here, in label column we can see that all Male is converted into 1 and all Female value is converted into 0.

1. **NORMALISING THE DATA:**

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information. Normalization is also required for some algorithms to model the data correctly.

For example, assume your input dataset contains one column with values ranging from 0 to 1, and another column with values ranging from 10,000 to 100,000. The great difference in the scale of the numbers could cause problems when you attempt to combine the values as features during modeling.

Normalization avoids these problems by creating new values that maintain the general distribution and ratios in the source data, while keeping values within a scale applied across all numeric columns used in the model.

Graphical user interface, application

Description automatically generated

Here we can see our data is normalized between 0 and 1 using MinMaxScaler.

1. **SPLITTING DATA INTO TEST AND TRAIN USING TEST\_TRAIN SPLIT:**

We performed splitting in our dataset in train and test. Ratio (train: test::0.7: 0.3) or train = 70% of data and test = 30% of data.

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, text

Description automatically generated**

1. **TRAINING MACHINE LEARNING MODEL:**

In this step, we trained our machine learning model based on the model we choose using IBM SPSS Modeler to achieve maximum accuracy.

Table

Description automatically generated

1. **EVALUATING OUR TRAINED MODEL ON TEST DATA:**

After training our ML model we tested our model on test data and evaluated its performance based on accuracy.

Background pattern

Description automatically generated

The above array is the prediction of our trained model on the test data. This array will be compared with actual/true value and the accuracy will be calculated.

Graphical user interface, text, application, email

Description automatically generated

Here, using accuracy score we can see that we got accuracy of 97.8%.

To check how many labels were mis predicted and correctly predicted by our ML model we will use confusion matrix.

Chart

Description automatically generated

Here we can conclude the following points:

* Our model predicted Females which were Females were 502/511. (Correctly predicted Females)
* Our model predicted Males which were Females were 009/511. (Mis prediction of Females)
* Our model predicted Female which were Males were 011/440. (Mis prediction of Males)
* Our model predicted Males which were Males were 429/440. (Correctly predicted Males).

1. **EXPORTING TRAINED MODEL FOR FRONT\_END:**

We will now export our ML model to use it in our front end in upcoming phase using pickle library.

Graphical user interface, text, application

Description automatically generated

***Conclusion:***

In phase-I we created and deployed the back end of the application/project and achieved brilliant accuracy of 97.8%. ANN (Artificial Neural Network) and Front end will be created in the Phase-II of the project.

The machine learning model has been built. Further improvements will be made to the model to make it fast and efficient.

The final step includes integration of both application UI and machine learning model which will be covered in Phase-II of the major project.

***Contributions:***

**Sparsh Saxena:**

* Worked on jupyter notebook to create machine learning model.
* Documentation (Report).
* Learned and worked on IBM SPSS Modeler.

**Shivani Deoli:**

* Decided the Front-End application workflow.
* UI development.
* Worked on different app functionalities.

***References:***

Dataset used to train model-:

<https://raw.githubusercontent.com/primaryobjects/voice-gender/master/voice.csv>

To Build concept of the project-:

<http://www.primaryobjects.com/2016/06/22/identifying-the-gender-of-a-voice-using-machine-learning/>