Name: Yuvraj Arora

Reg-no: 11814537

Subject: Int248

CA 1

Gender Recognition Using Voice

Introduction

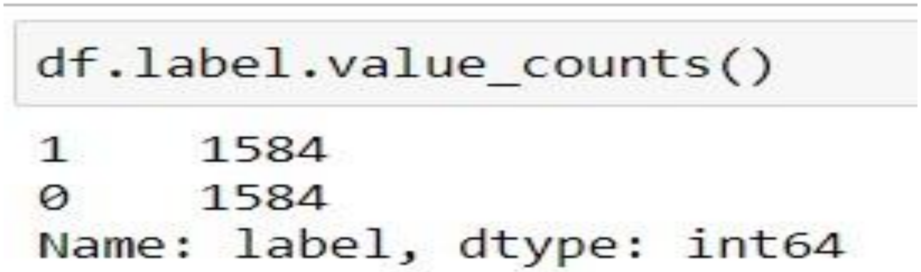
The Main goal is to classify Gender using voice from the given dataset. Gender classification using voice is simple. Gender can be differentiated from voice based on pitch of voice. Female voice have high pitch as compared to Male voice. And some other factors can also be used to classify Gender. The dataset used is voice.csv which was imported from Kaggle. The goal is to classify the gender based on the voice features. The code was written in Python 3. Python 3 environment comes with many helpful analytics libraries installed. The data then was visualized using SNS plot, Histogram, SNSHeatmap which makes it easier to visualize data.

Dataset:

The dataset used was imported from Kaggle

Dataset Description: The goal is to classify the gender based on the voice features

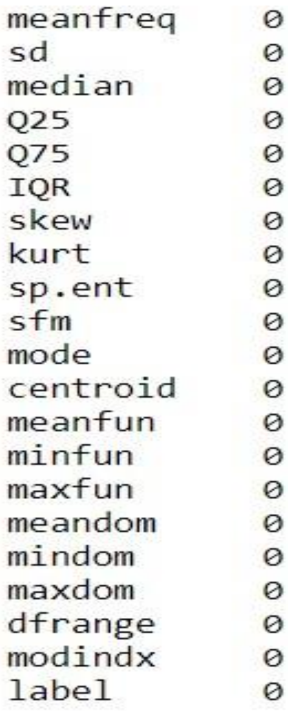
The dataset contains 21 columns including 20 attributes and 1 target values which is the gender The Dimensions of the dataset are 3167x21



```
df.label.value_counts()
```

```
1     1584
0     1584
Name: label, dtype: int64
```

df.isnull().sum()



```
meanfreq    0
sd          0
median      0
Q25         0
Q75         0
IQR         0
skew        0
kurt        0
sp.ent      0
sfm         0
mode        0
centroid    0
meanfun     0
minfun      0
maxfun      0
meandom     0
mindom      0
maxdom      0
dfrange     0
modindx     0
label       0
```

Import Libraries

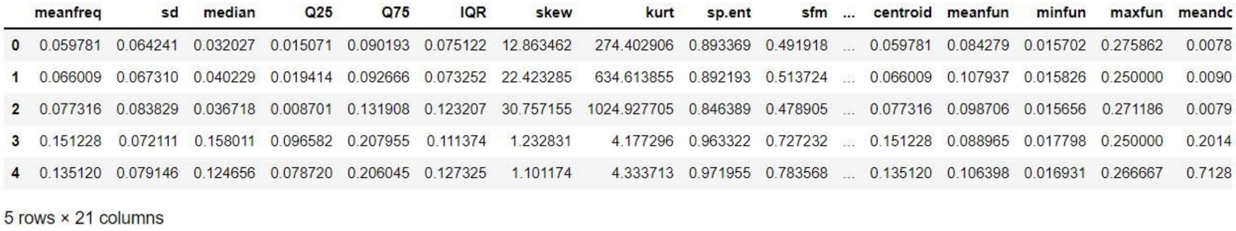
- import pandas as pd
- import numpy as np
- import matplotlib.pyplot as plt
- import seaborn as sns

Load the Data

```
from google.colab import files


uploaded = files.upload()
```

```
df=pd.read_csv('voice.csv')
```

```
df.head()
```



	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt	sp.ent	sfm	...	centroid	meanfun	minfun	maxfun	meandc
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	0.0078
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	0.0090
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	0.0079
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	0.2014
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	0.7128

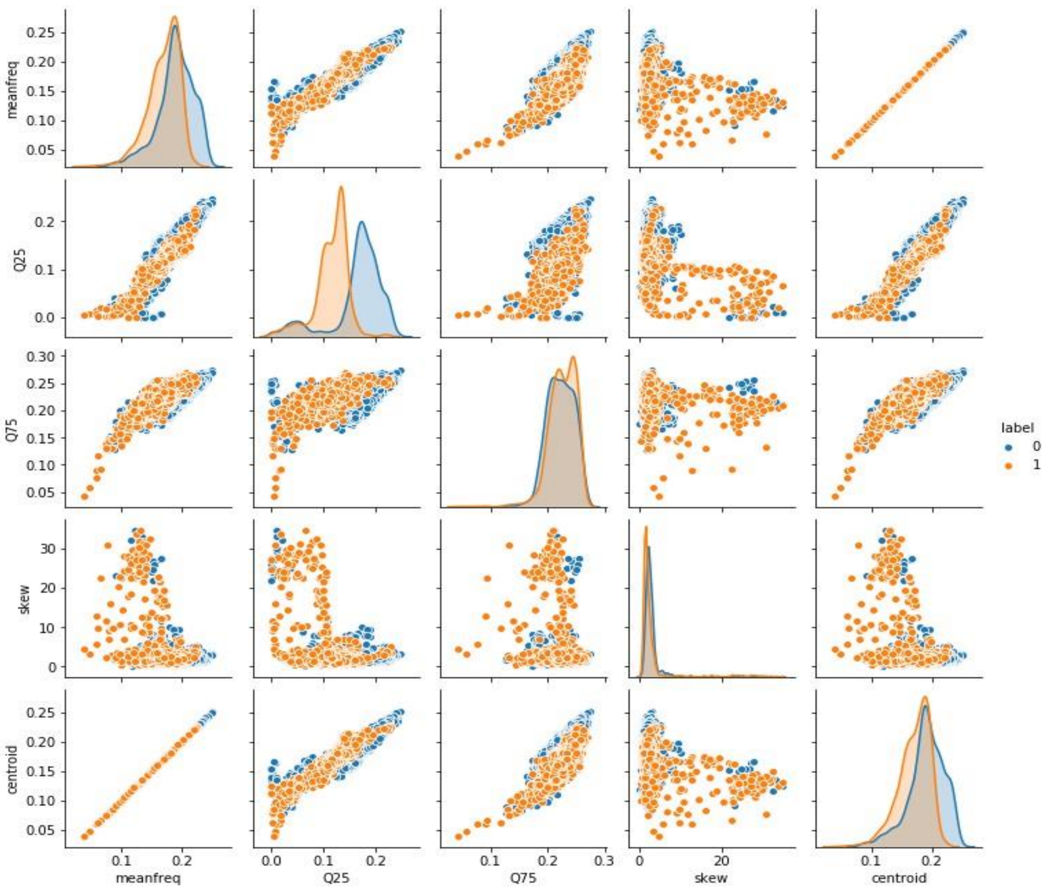
5 rows × 21 columns

Data Visualization

- sns.pairplot(df[['meanfreq', 'Q25', 'Q75', 'skew', 'centroid', 'label']], hue='label', size=2)

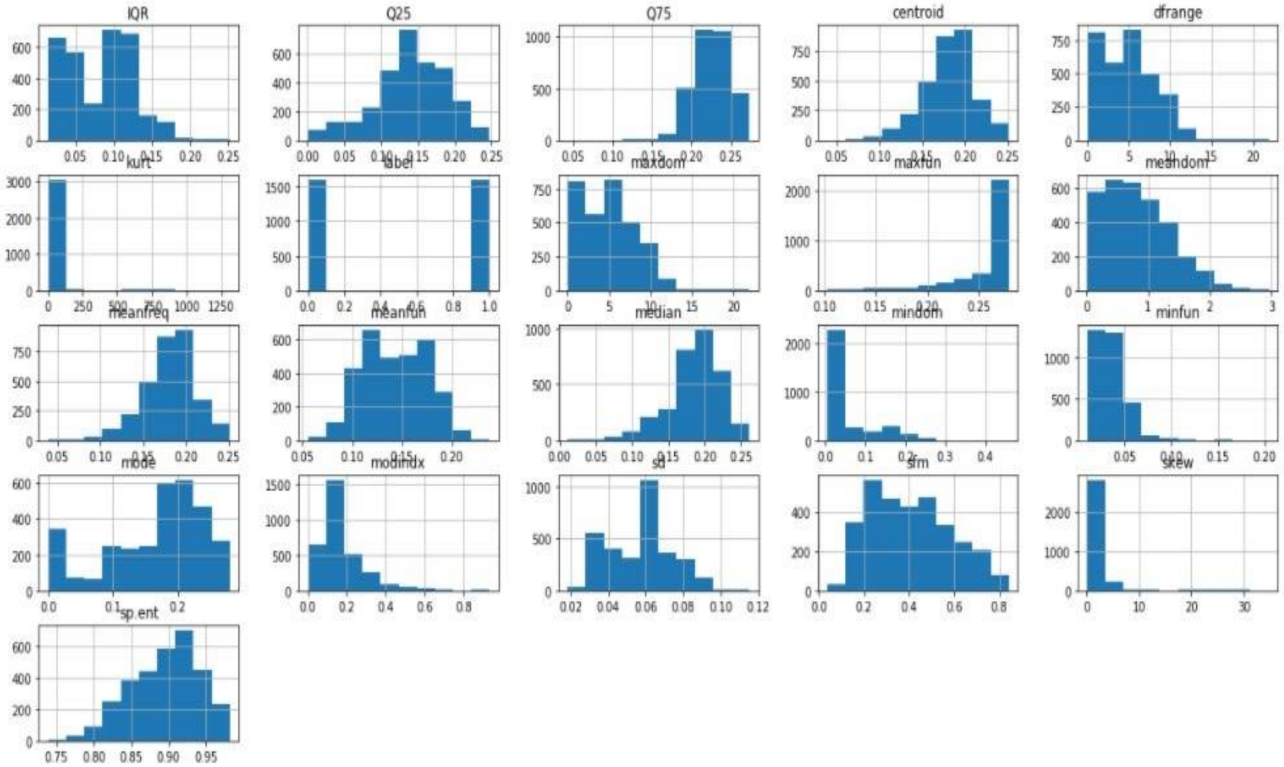
- plt.show()

This pairplot is plotted to show the relations between the dataset features and we find that the skewness with the quartiles are having a weak relation unlike the meanfreq withthe centroid as they have a strong relation.



Histogram

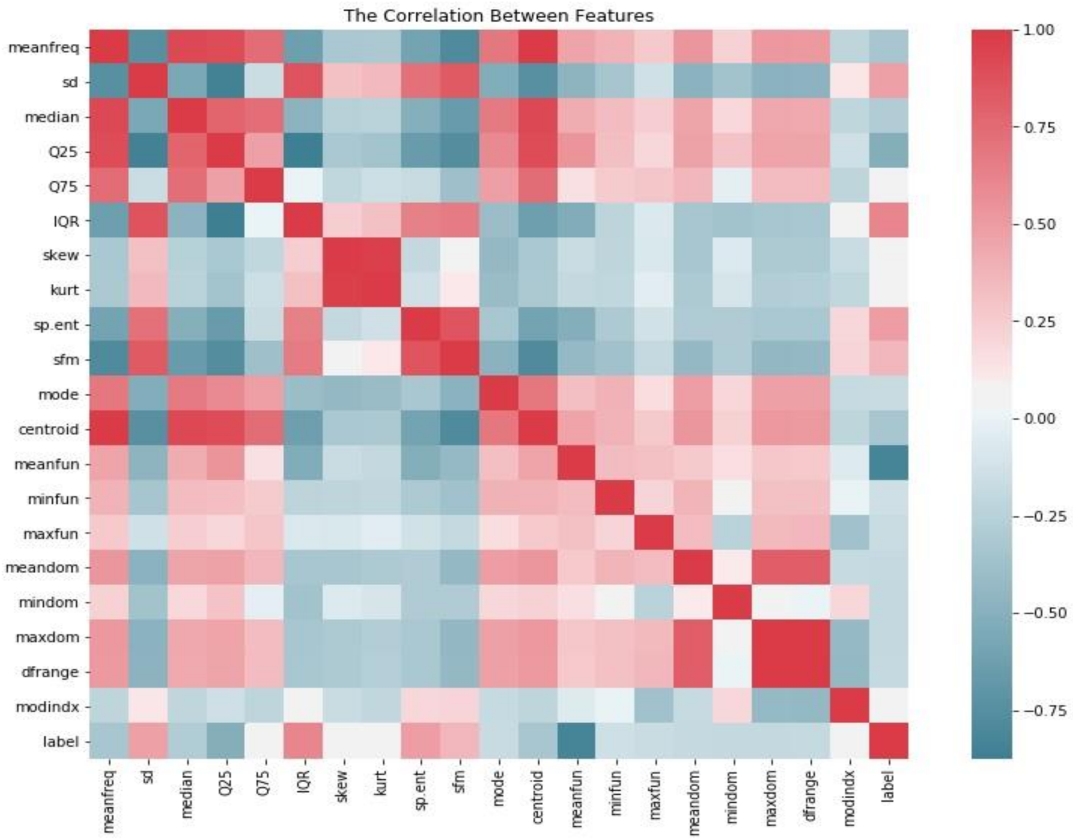
- df.hist(figsize=(21, 10))
- plt.show()



Check the Correlation

```
◦ f, ax = plt.subplots(figsize=(15, 10))
◦ corr = df.corr()
◦ sns.heatmap(corr,mask=np.zeros_like(corr, dtype=np.bool), cmap=sns.diverging_palette(220, 10, as_cmap=True),square=True, ax=ax)
◦ plt.title('The Correlation Between Features')
◦ plt.show()
```

This Heatmap shows the correlations between the features; and by looking at it we find a positive correlation between the label and IQR, label and sp.ent, and a huge negative correlation between the label and meanfun



PCA

◦ from sklearn.preprocessing import scale

◦ from sklearn.preprocessing import StandardScaler

◦ from sklearn.decomposition import PCA

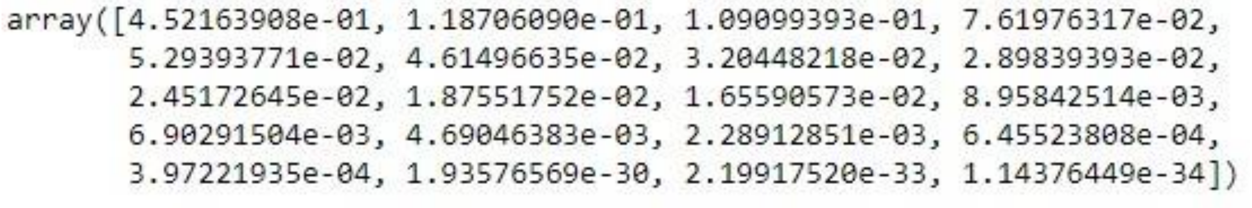
Scaling the values

#The amount of variance that each PC explains

var= pca.explained_variance_ratio_ var

x = scale(X)

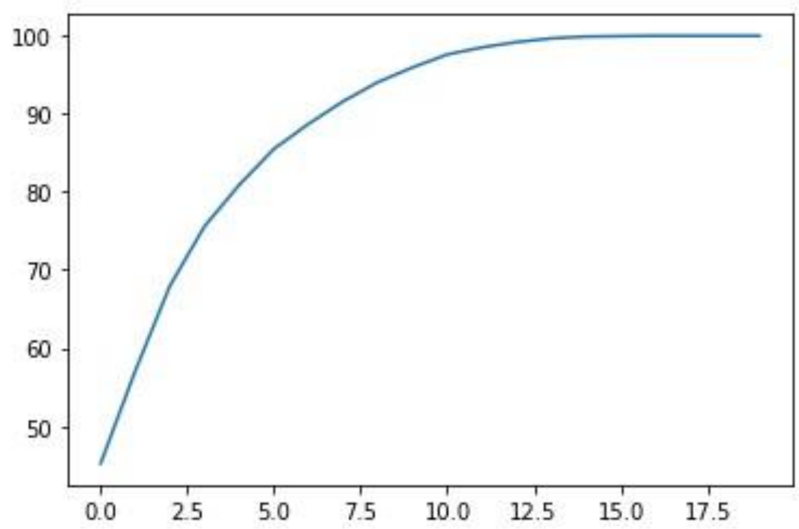
```
var= pca.explained_variance_ratio_

var
```



```
array([4.52163908e-01, 1.18706090e-01, 1.09099393e-01, 7.61976317e-02,
       5.29393771e-02, 4.61496635e-02, 3.20448218e-02, 2.89839393e-02,
       2.45172645e-02, 1.87551752e-02, 1.65590573e-02, 8.95842514e-03,
       6.90291504e-03, 4.69046383e-03, 2.28912851e-03, 6.45523808e-04,
       3.97221935e-04, 1.93576569e-30, 2.19917520e-33, 1.14376449e-34])
```

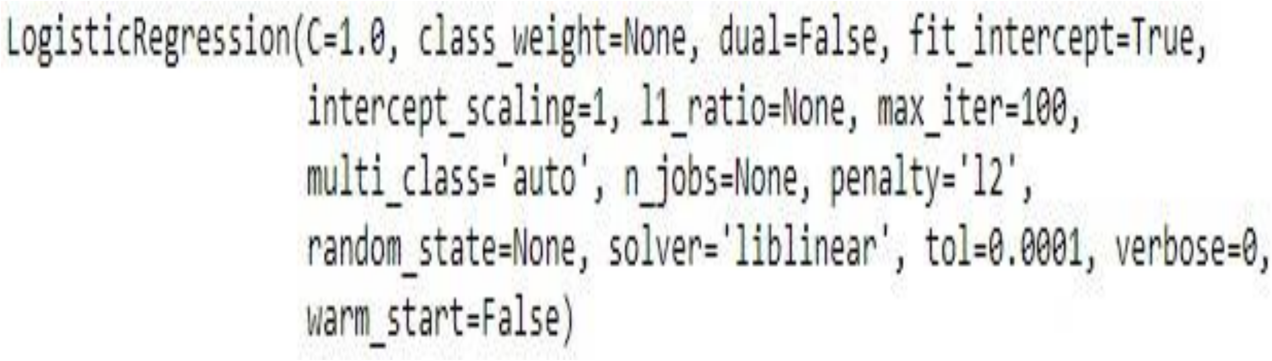
PCA

```
var1=np.cumsum(np.round(pca.explained_variance_ratio_, decimals=4)*100)
var1
plt.plot(var1)
```



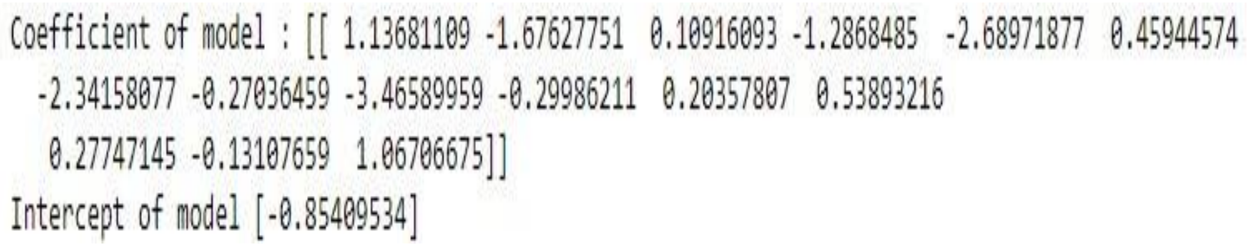
Model Building using Logistic regression

- model= LogisticRegression(solver='liblinear')
- model.fit(x_train,y_train)



```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='liblinear', tol=0.0001, verbose=0,
                   warm_start=False)
```

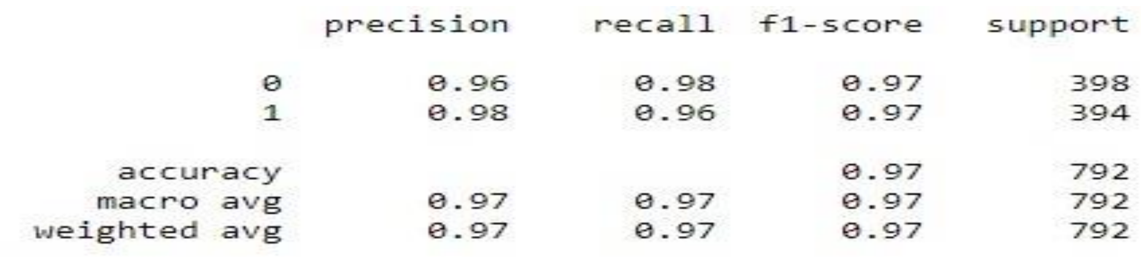
```
print('Coefficient of model :', model.coef_)
print('Intercept of model',model.intercept_)
```



```
Coefficient of model : [[ 1.13681109 -1.67627751  0.10916093 -1.2868485  -2.68971877  0.45944574
  -2.34158077 -0.27036459 -3.46589959 -0.29986211  0.20357807  0.53893216
   0.27747145 -0.13107659  1.06706675]]
Intercept of model [-0.85409534]
```

Model Building using Logistic regression

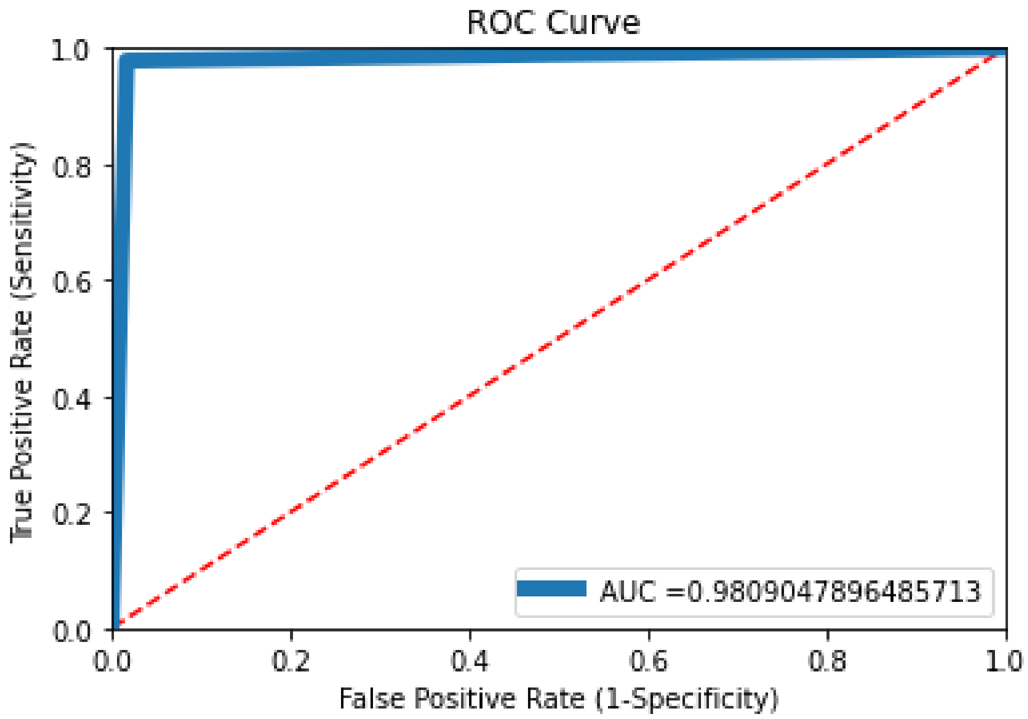
```
ypred = model.predict(x_test)
from sklearn.metrics import classification_report, confusion_matrix
print(classification_report(y_test,ypred))
```



```
              precision    recall  f1-score   support

           0       0.96      0.98      0.97       398
           1       0.98      0.96      0.97       394

    accuracy                           0.97       792
   macro avg       0.97      0.97      0.97       792
weighted avg       0.97      0.97      0.97       792
```

Roc Curve

```
plt.title('ROC Curve ')
plt.xlabel('False Positive Rate (1-Specificity)')
plt.ylabel('True Positive Rate (Sensitivity)')
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.plot(fpr, tpr, label = 'AUC =' + str(roc_auc), lw=6)
plt.legend(loc=4) #Location of label
plt.show()
```



Conclusion:

In the next 5-10 years, it’s highly unlikely machine driven speech-to-text won’t surpass human transcription in both accuracy and speed, just given the current pace of development. We’re not there yet, but we will be shortly. That speed of development will also increase over the next few years as we continue to capture more voice data through in-home and mobile virtual assistant apps like Siri and Alexa. So it is clear that voice recognition has greater scope in future. And due to which Gender recognition using Voice will be necessary and will be very useful in the future.

References :

DATASET

Taken From Internet has record of voices of different people of different Gender.

SNSPLOT

https://seaborn.pydata.org/generated/seaborn.lineplot.html

LOGISTIC REGRESSION

https://en.wikipedia.org/wiki/Logistic_regression

THANKYOU