

```
#logistic regression- preametric algorithm means it assumption about data
#NAIVE BAYS -
#DECISION TREE - NON PARAMETRIC ; NOT TAKE ASSUMPTION ON DATA
```

```
#HOW TO CHECK NORMALITY
#GRAPHS,Q-Q PLOT ,HISTOGRAM,KDE PLOT ,ks TEST - ANALYTICAL METHOD
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, roc_curve, roc_auc_score
```

```
df = pd.read_csv('/content/Gender_voice_Classification.csv')
```

```
print(df.info())
print(df.head())
```

```
df['Gender'] = LabelEncoder().fit_transform(df['Gender']) # Male = 1, Female = 0
```

```
X = df.drop(columns=['Gender'])
y = df['Gender']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
model = DecisionTreeClassifier(random_state=5, max_depth=3, criterion='gini')
model.fit(X_train, y_train)
```

```
y_pred_train = model.predict(X_train)
y_pred_test = model.predict(X_test)
```

```
print(f'Training Accuracy: {accuracy_score(y_train, y_pred_train)}')
print(f'Testing Accuracy: {accuracy_score(y_test, y_pred_test)}')
```

```
print("Classification Report (Train):\n", classification_report(y_train, y_pred_train))
print("Classification Report (Test):\n", classification_report(y_test, y_pred_test))
```

```
cm_train = confusion_matrix(y_train, y_pred_train)
cm_test = confusion_matrix(y_test, y_pred_test)
```

```
plt.figure(figsize=(5,4))
sns.heatmap(cm_train, annot=True, cmap='Blues', fmt='d')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix - Training')
plt.show()
```

```
plt.figure(figsize=(5,4))
sns.heatmap(cm_test, annot=True, cmap='Oranges', fmt='d')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix - Testing')
plt.show()
```

```
y_pred_prob = model.predict_proba(X_test)[:,-1]
fpr, tpr, _ = roc_curve(y_test, y_pred_prob)
auc_score = roc_auc_score(y_test, y_pred_prob)
```

```
plt.figure(figsize=(6,5))
plt.plot(fpr, tpr, label=f'AUC = {auc_score:.2f}', color='blue')
plt.plot([0,1], [0,1], linestyle='--', color='gray')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend()
plt.show()
```

```
plt.figure(figsize=(12,8))
plot_tree(model, feature_names=X.columns, class_names=['Female', 'Male'], filled=True, rounded=True)
plt.show()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3168 entries, 0 to 3167
Data columns (total 21 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   meanfreq    3168 non-null   float64
 1   sd          3168 non-null   float64
 2   median      3168 non-null   float64
 3   Q25         3168 non-null   float64
 4   Q75         3168 non-null   float64
 5   IQR         3168 non-null   float64
 6   skew        3168 non-null   float64
 7   kurt        3168 non-null   float64
 8   sp.ent      3168 non-null   float64
 9   sfm         3168 non-null   float64
10   mode        3168 non-null   float64
11   centroid    3168 non-null   float64
12   meanfun     3168 non-null   float64
13   minfun      3168 non-null   float64
14   maxfun      3168 non-null   float64
15   meandom     3168 non-null   float64
16   mindom      3168 non-null   float64
17   maxdom      3168 non-null   float64
18   dfrange     3168 non-null   float64
19   modindx     3168 non-null   float64
20   Gender      3168 non-null   object
dtypes: float64(20), object(1)
memory usage: 519.9+ KB
None
   meanfreq      sd      median      Q25      Q75      IQR      skew \
0  0.059781  0.064241  0.032027  0.015071  0.090193  0.075122  12.863462
1  0.066009  0.067310  0.040229  0.019414  0.092666  0.073252  22.423285
2  0.077316  0.083829  0.036718  0.008701  0.131908  0.123207  30.757155
3  0.151228  0.072111  0.158011  0.096582  0.207955  0.111374   1.232831
4  0.135120  0.079146  0.124656  0.078720  0.206045  0.127325   1.101174

   kurt      sp.ent      sfm      ...  centroid  meanfun  minfun \
0  274.402905  0.893369  0.491918  ...  0.059781  0.084279  0.015702
1  634.613855  0.892193  0.513724  ...  0.066009  0.107937  0.015826
2  1024.927705  0.846389  0.478905  ...  0.077316  0.098706  0.015656
3    4.177296  0.963322  0.727232  ...  0.151228  0.088965  0.017798
4    4.333713  0.971955  0.783568  ...  0.135120  0.106398  0.016931

   maxfun  meandom  mindom  maxdom  dfrange  modindx  Gender
0  0.275862  0.007812  0.007812  0.007812  0.000000  0.000000  male
1  0.250000  0.009014  0.007812  0.054688  0.046875  0.052632  male
2  0.271186  0.007990  0.007812  0.015625  0.007812  0.046512  male
3  0.250000  0.201497  0.007812  0.562500  0.554688  0.247119  male
4  0.266667  0.712812  0.007812  5.484375  5.476562  0.208274  male

[5 rows x 21 columns]
Training Accuracy: 0.970681100586378
Testing Accuracy: 0.964248159831756
Classification Report (Train):
      precision    recall  f1-score   support

     0       0.98       0.96       0.97       1132
     1       0.96       0.98       0.97       1085

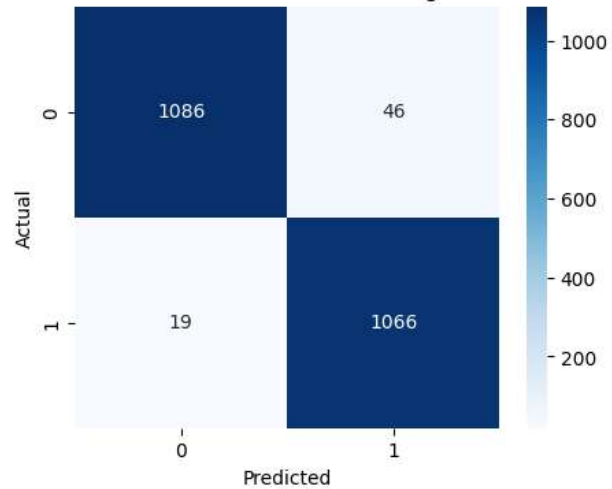
   accuracy                0.97       2217
  macro avg              0.97       0.97       0.97       2217
 weighted avg              0.97       0.97       0.97       2217

```

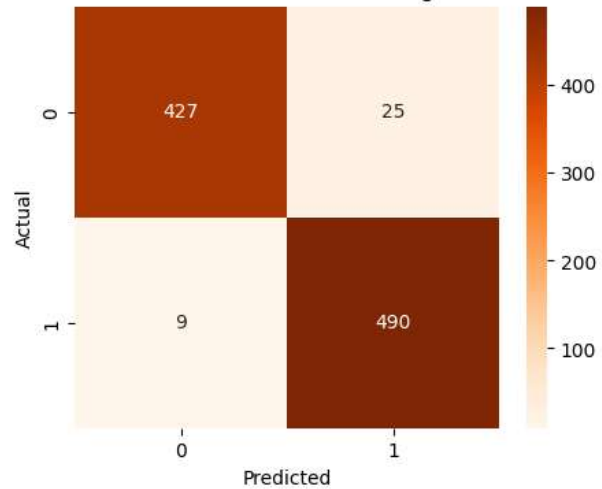
Classification Report (Test):

	precision	recall	f1-score	support
0	0.98	0.94	0.96	452
1	0.95	0.98	0.97	499
accuracy			0.96	951
macro avg	0.97	0.96	0.96	951
weighted avg	0.96	0.96	0.96	951

Confusion Matrix - Training



Confusion Matrix - Testing



ROC Curve



