

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

In [3]: df = pd.read_csv('Voice.csv')
df.head(18)

Out[3]:
```

|   | meanfreq | sd       | median   | Q25      | Q75      | IQR      | skew      | kurt        | sp.ent   | sfn      | ... | centroid | meanfun  | minfun   | maxfun   | meandom  | mindom   | maxdom   | dfrange  | mod  |
|---|----------|----------|----------|----------|----------|----------|-----------|-------------|----------|----------|-----|----------|----------|----------|----------|----------|----------|----------|----------|------|
| 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 | 0.007812 | 0.007812 | 0.007812 | 0.000000 | 0.00 |
| 1 | 0.066009 | 0.067310 | 0.040229 | 0.018414 | 0.092666 | 0.073252 | 22.423285 | 634.613855  | 0.892193 | 0.513724 | ... | 0.066009 | 0.107937 | 0.015826 | 0.250000 | 0.009014 | 0.007812 | 0.054688 | 0.046875 | 0.05 |
| 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 | 0.007990 | 0.007812 | 0.015625 | 0.007812 | 0.04 |
| 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 | 0.201497 | 0.007812 | 0.562500 | 0.554688 | 0.24 |
| 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 | 0.712812 | 0.007812 | 5.484375 | 5.476562 | 0.20 |
| 5 | 0.132786 | 0.079557 | 0.119090 | 0.067958 | 0.209592 | 0.141634 | 1.932562  | 8.308895    | 0.963181 | 0.738307 | ... | 0.132786 | 0.110132 | 0.017112 | 0.253968 | 0.298222 | 0.007812 | 2.726562 | 2.718750 | 0.12 |
| 6 | 0.150762 | 0.074463 | 0.160106 | 0.092899 | 0.205718 | 0.112619 | 1.530643  | 5.987498    | 0.967573 | 0.762638 | ... | 0.150762 | 0.105945 | 0.026230 | 0.266667 | 0.479620 | 0.007812 | 5.312500 | 5.304688 | 0.12 |
| 7 | 0.160514 | 0.076767 | 0.144337 | 0.110532 | 0.231962 | 0.121430 | 1.397156  | 4.766611    | 0.959255 | 0.719858 | ... | 0.160514 | 0.093052 | 0.017758 | 0.144144 | 0.301339 | 0.007812 | 0.539062 | 0.531250 | 0.28 |
| 8 | 0.142239 | 0.078018 | 0.138587 | 0.088206 | 0.208587 | 0.120381 | 1.099746  | 4.070284    | 0.970723 | 0.770992 | ... | 0.142239 | 0.096729 | 0.017957 | 0.250000 | 0.336476 | 0.007812 | 2.164062 | 2.156250 | 0.14 |
| 9 | 0.134329 | 0.080350 | 0.121451 | 0.075580 | 0.201957 | 0.126377 | 1.190368  | 4.787310    | 0.975246 | 0.804505 | ... | 0.134329 | 0.105881 | 0.019300 | 0.262295 | 0.340365 | 0.015625 | 4.695312 | 4.679688 | 0.08 |

10 rows × 21 columns

```
In [4]: df.shape
Out[4]: (3168, 21)

In [5]: df.isnull().sum()

Out[5]:
```

|          | meanfreq | sd | median | Q25 | Q75 | IQR | skew | kurt | sp.ent | sfn | mode | centroid | meanfun | minfun | maxfun | meandom | mindom | maxdom | dfrange | mod |
|----------|----------|----|--------|-----|-----|-----|------|------|--------|-----|------|----------|---------|--------|--------|---------|--------|--------|---------|-----|
| meanfreq | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| sd       | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| median   | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| Q25      | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| Q75      | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| IQR      | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| skew     | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| kurt     | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| sp.ent   | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| sfn      | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| mode     | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| centroid | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| meanfun  | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| minfun   | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| maxfun   | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| meandom  | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| mindom   | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| maxdom   | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| dfrange  | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| modindx  | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| label    | 0        |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |
| dtype:   | int64    |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |     |

```
In [6]: df.dtypes

Out[6]:
```

|          | meanfreq | sd | median | Q25 | Q75 | IQR | skew | kurt | sp.ent | sfn | mode | centroid | meanfun | minfun | maxfun | meandom | mindom | maxdom | dfrange | modindx | label | dtype: |
|----------|----------|----|--------|-----|-----|-----|------|------|--------|-----|------|----------|---------|--------|--------|---------|--------|--------|---------|---------|-------|--------|
| meanfreq | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| sd       | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| median   | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| Q25      | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| Q75      | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| IQR      | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| skew     | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| kurt     | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| sp.ent   | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| sfn      | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| mode     | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| centroid | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| meanfun  | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| minfun   | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| maxfun   | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| meandom  | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| mindom   | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| maxdom   | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| dfrange  | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| modindx  | float64  |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| label    | object   |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |
| dtype:   | object   |    |        |     |     |     |      |      |        |     |      |          |         |        |        |         |        |        |         |         |       |        |

```
In [7]: df['label']

Out[7]:
```

|      | label  |
|------|--------|
| 0    | male   |
| 1    | male   |
| 2    | male   |
| 3    | male   |
| 4    | male   |
| ...  | ...    |
| 3163 | female |
| 3164 | female |
| 3165 | female |
| 3166 | female |
| 3167 | female |

Name: label, Length: 3168, dtype: object

```
In [8]: df['label'].value_counts()

Out[8]:
```

| label  | count |
|--------|-------|
| male   | 1584  |
| female | 1584  |

Name: label, dtype: int64

```
In [9]: from sklearn.preprocessing import LabelEncoder

In [10]: lb = LabelEncoder()
df['label'] = lb.fit_transform(df['label'])

In [11]: df['label']

Out[11]:
```

|      | label |
|------|-------|
| 0    | 1     |
| 1    | 1     |
| 2    | 1     |
| 3    | 1     |
| 4    | 1     |
| ...  | ...   |
| 3163 | 0     |
| 3164 | 0     |
| 3165 | 0     |
| 3166 | 0     |
| 3167 | 0     |

Name: label, Length: 3168, dtype: int32

```
In [12]: df['label'].value_counts()

Out[12]:
```

| label | count |
|-------|-------|
| 1     | 1584  |
| 0     | 1584  |

Name: label, dtype: int64

```
In [13]: df.columns

Out[13]:
```

Index(['meanfreq', 'sd', 'median', 'Q25', 'Q75', 'IQR', 'skew', 'kurt', 'sp.ent', 'sfn', 'mode', 'centroid', 'meanfun', 'minfun', 'maxfun', 'meandom', 'mindom', 'maxdom', 'dfrange', 'modindx', 'label'], dtype='object')

```
In [14]: df['meanfreq'].value_counts()

Out[14]:
```

| meanfreq | count |
|----------|-------|
| 0.212190 | 2     |
| 0.213732 | 2     |
| 0.059781 | 1     |
| 0.188687 | 1     |
| 0.191174 | 1     |
| ...      | ...   |
| 0.192408 | 1     |
| 0.192558 | 1     |
| 0.194565 | 1     |
| 0.196195 | 1     |
| 0.165569 | 1     |

Name: meanfreq, Length: 3166, dtype: int64

```
In [15]: df['sd'].value_counts()

Out[15]:
```

| sd       | count |
|----------|-------|
| 0.043190 | 2     |
| 0.057785 | 2     |
| 0.064241 | 1     |
| 0.030689 | 1     |
| 0.633622 | 1     |
| ...      | ...   |
| 0.055248 | 1     |
| 0.055987 | 1     |
| 0.067716 | 1     |
| 0.059108 | 1     |
| 0.092884 | 1     |

Name: sd, Length: 3166, dtype: int64

```
In [16]: df['median'].value_counts()

Out[16]:
```

| median   | count |
|----------|-------|
| 0.186667 | 6     |
| 0.220600 | 4     |
| 0.172832 | 3     |
| 0.183448 | 3     |
| 0.179208 | 3     |
| ...      | ...   |
| 0.202273 | 1     |
| 0.203671 | 1     |
| 0.205660 | 1     |
| 0.218593 | 1     |
| 0.183644 | 1     |

Name: median, Length: 3077, dtype: int64

```
In [17]: df['mindom'].value_counts()

Out[17]:
```

| mindom   | count |
|----------|-------|
| 0.023438 | 3246  |
| 0.007812 | 814   |
| 0.164662 | 109   |
| 0.054688 | 63    |
| 0.004883 | 61    |
| ...      | ...   |
| 0.458984 | 1     |
| 0.351562 | 1     |
| 0.027344 | 1     |
| 0.619531 | 1     |
| 0.107666 | 1     |

Name: mindom, Length: 77, dtype: int64

```
In [18]: plt.figure(figsize=(10,8))
plt.pie(df['label'].value_counts(), labels=df['label'].value_counts().index, autopct='%2F%%')
plt.show()
```

```
In [19]: from sklearn.model_selection import train_test_split

In [23]: x = df.drop('label',axis=1)
y = df['label']
print(type(x))
print(type(y))
print(x.shape)
print(y.shape)

<class 'pandas.core.frame.DataFrame'>
<class 'pandas.core.series.Series'>
(3168, 20)
(3168,)

In [21]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.20)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)

(2534, 20)
(634, 20)
(2534,)
(634,)
```

### Model Building

```
In [40]: from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score

In [41]: def gen_cls_metrics(ytest,ypred):
    print('Accuracy Score',accuracy_score(ytest,ypred))
    cm = confusion_matrix(ytest,ypred)
    print(cm)
    print(classification_report(ytest,ypred))

    def train_test_score(model):
        print('Training Score',model.score(x_train,y_train))
        print('Testing Score',model.score(x_test,y_test))

1) DecisionTree Classifier

In [42]: print(x_train.shape)
print(y_train.shape)

(2534, 20)
(2534,)
```

```
In [43]: m1 = DecisionTreeClassifier(criterion='gini',max_depth=5)
m1.fit(x_train,y_train)

Out[43]: DecisionTreeClassifier(max_depth=5)

In [44]: train_test_score(m1)

Training Score 0.9846093133385951
Testing Score 0.9684542586750788

In [45]: ypred_m1 = m1.predict(x_test)

In [46]: print('Metrics for Decistion Tree Classifier')
gen_cls_metrics(y_test,ypred_m1)

Metrics for Decistion Tree Classifier
Accuracy Score 0.9684542586750788
[[330  7]
 [ 13 284]]
precision    recall  f1-score   support

   0         0.96         0.98         0.97         337
   1         0.98         0.96         0.97         297

 accuracy         0.97         0.97         0.97         634
 macro avg         0.97         0.97         0.97         634
weighted avg         0.97         0.97         0.97         634

2) RandomForest Classifier

In [47]: m2 = RandomForestClassifier(n_estimators=70,criterion='gini',max_depth=5)
m2.fit(x_train,y_train)

Out[47]: RandomForestClassifier(max_depth=5, n_estimators=70)

In [48]: train_test_score(m2)

Training Score 0.9834254143646409
Testing Score 0.9716088328075709

In [49]: ypred_m2 = m2.predict(x_test)

In [50]: print('Metrics for random Forest Classifier')
gen_cls_metrics(y_test,ypred_m2)

Metrics for random Forest Classifier
Accuracy Score 0.9716088328075709
[[328  9]
 [  9 288]]
precision    recall  f1-score   support

   0         0.97         0.97         0.97         337
   1         0.97         0.97         0.97         297

 accuracy         0.97         0.97         0.97         634
 macro avg         0.97         0.97         0.97         634
weighted avg         0.97         0.97         0.97         634

3) KNN

In [51]: m3 = KNeighborsClassifier(n_neighbors=23)
m3.fit(x_train,y_train)

Out[51]: KNeighborsClassifier(n_neighbors=23)

In [52]: train_test_score(m3)

Training Score 0.7328334648776638
Testing Score 0.6829652996845426

In [53]: ypred_m3 = m3.predict(x_test)

In [54]: print('Metrics for KNN Classifier')
gen_cls_metrics(y_test,ypred_m3)

Metrics for KNN Classifier
Accuracy Score 0.6829652996845426
[[229 108]
 [ 93 284]]
precision    recall  f1-score   support

   0         0.71         0.68         0.69         337
   1         0.65         0.69         0.67         297

 accuracy         0.68         0.68         0.68         634
 macro avg         0.68         0.68         0.68         634
weighted avg         0.68         0.68         0.68         634

4) LogisticRegression

In [56]: m4 = LogisticRegression(max_iter = 1000)
m4.fit(x_train,y_train)

Out[56]: LogisticRegression(max_iter=1000)

In [57]: train_test_score(m4)

Training Score 0.9100236779794791
Testing Score 0.9085173501577287

In [58]: ypred_m4 = m4.predict(x_test)

In [59]: print('Metrics for Log_Reg Classifier')
gen_cls_metrics(y_test,ypred_m4)

Metrics for Log_Reg Classifier
Accuracy Score 0.9085173501577287
[[289  48]
 [ 10 287]]
precision    recall  f1-score   support

   0         0.86         0.86         0.91         337
   1         0.97         0.97         0.91         297

 accuracy         0.91         0.91         0.91         634
 macro avg         0.91         0.91         0.91         634
weighted avg         0.92         0.91         0.91         634

5) SVM

In [60]: m5 = SVC(kernel='linear',C=0.1)
m5.fit(x_train,y_train)

Out[60]: SVC(C=0.1, kernel='linear')

In [61]: train_test_score(m5)

Training Score 0.82872992817679558
Testing Score 0.807570977917981

In [63]: ypred_m5 = m5.predict(x_test)

In [64]: print('Metrics for SVM Classifier')
gen_cls_metrics(y_test,ypred_m5)

Metrics for SVM Classifier
Accuracy Score 0.807570977917981
[[251  86]
 [ 36 261]]
precision    recall  f1-score   support

   0         0.87         0.74         0.80         337
   1         0.75         0.88         0.81         297

 accuracy         0.81         0.81         0.81         634
 macro avg         0.81         0.81         0.81         634
weighted avg         0.82         0.81         0.81         634

Conclusion

Random Forest Classifier model has the highest accuracy of 97.16%

And the output on the test data set is as follows

where 1 represents male and 2 represents female

In [66]: print(ypred_m2)

[0 1 1 1 0 0 1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0
 1 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 0 0
 1 1 0 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1 1 1 1
 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 1
 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1
 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0
 1 0 0 0 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 1 0 0 0 0
```