```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import os
from sklearn.model_selection import train_test_split
```

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

	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt
0	0.059781	0.064241	0.032027	0.015071	0.090193	0.075122	12.863462	274.402906
1	0.066009	0.067310	0.040229	0.019414	0.092666	0.073252	22.423285	634.613855
2	0.077316	0.083829	0.036718	0.008701	0.131908	0.123207	30.757155	1024.927705
3	0.151228	0.072111	0.158011	0.096582	0.207955	0.111374	1.232831	4.177296
4	0.135120	0.079146	0.124656	0.078720	0.206045	0.127325	1.101174	4.333713

5 rows × 21 columns

```
df.info()
```

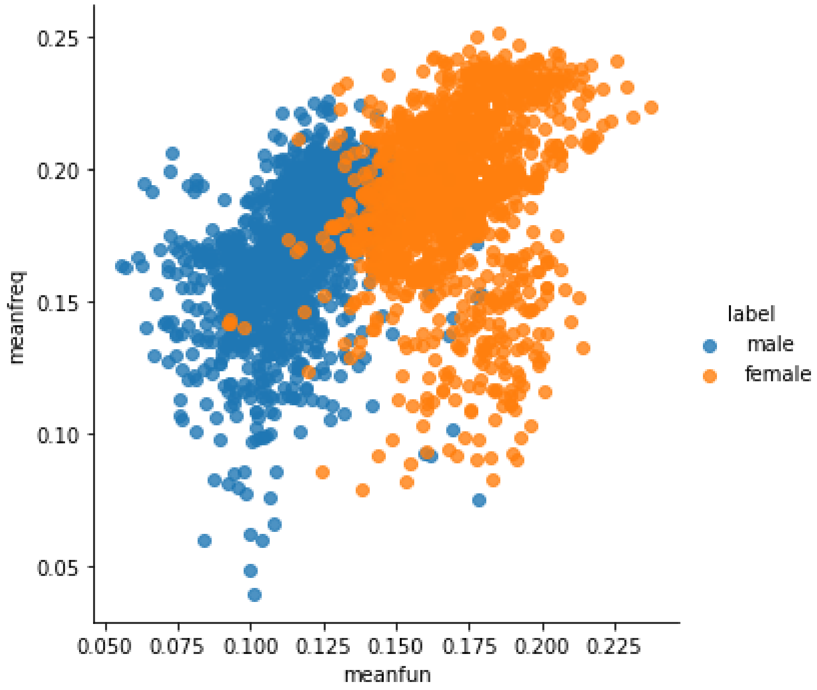
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3168 entries, 0 to 3167
Data columns (total 21 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   meanfreq  3168 non-null   float64
 1   sd        3168 non-null   float64
 2   median    3168 non-null   float64
 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  label     3168 non-null   object 
dtypes: float64(20), object(1)
memory usage: 519.9+ KB
```

```
df.columns
```

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

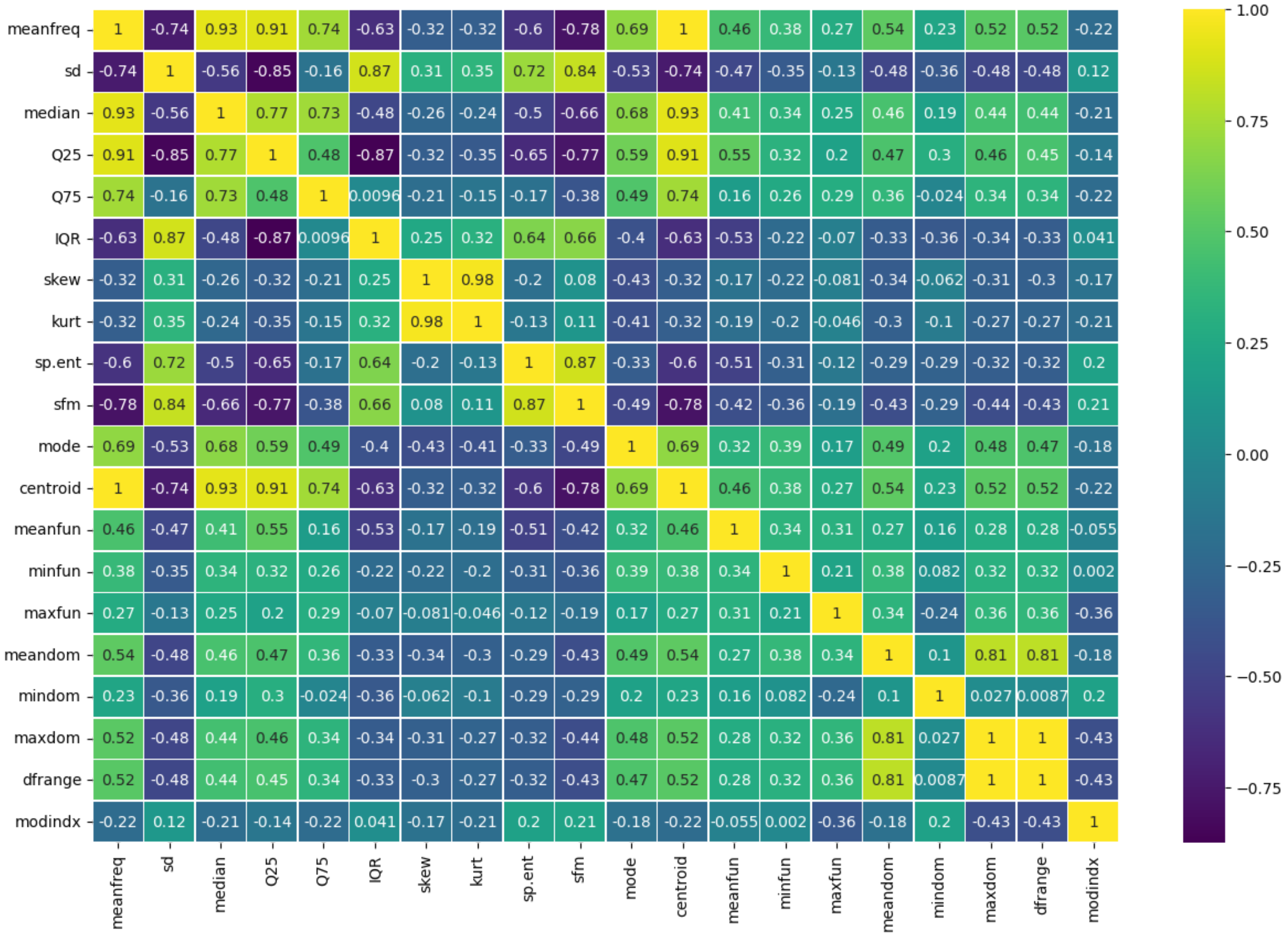
```
sns.lmplot(x='meanfun',y='meanfreq',hue='label',data=df,fit_reg=False)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f62dac44b50>
```



```
plt.figure(figsize=(15,10),dpi=100)
sns.heatmap(df.corr(),cmap="viridis",annot=True,linewidth=0.5)
```

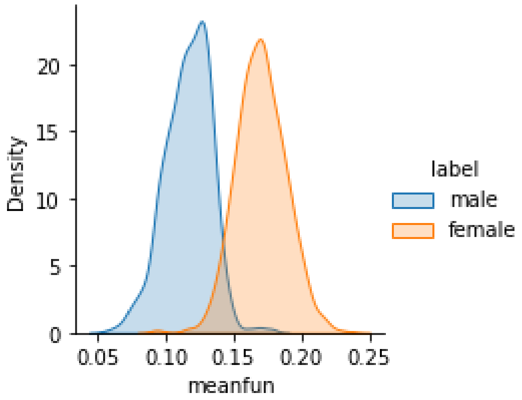
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f62d420e610>
```



	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt	sp.ent	sfm	mode	centroid	meanfun	minfun	maxfun	meandom	mindom	maxdom	dfrange	modindx
meanfreq	1	-0.74	0.93	0.91	0.74	-0.63	-0.32	-0.32	-0.6	-0.78	0.69	1	0.46	0.38	0.27	0.54	0.23	0.52	0.52	-0.22
sd	-0.74	1	-0.56	-0.85	-0.16	0.87	0.31	0.35	0.72	0.84	-0.53	-0.74	-0.47	-0.35	-0.13	-0.48	-0.36	-0.48	-0.48	0.12
median	0.93	-0.56	1	0.77	0.73	-0.48	-0.26	-0.24	-0.5	-0.66	0.68	0.93	0.41	0.34	0.25	0.46	0.19	0.44	0.44	-0.21
Q25	0.91	-0.85	0.77	1	0.48	-0.87	-0.32	-0.35	-0.65	-0.77	0.59	0.91	0.55	0.32	0.2	0.47	0.3	0.46	0.45	-0.14
Q75	0.74	-0.16	0.73	0.48	1	0.0096	-0.21	-0.15	-0.17	-0.38	0.49	0.74	0.16	0.26	0.29	0.36	-0.024	0.34	0.34	-0.22
IQR	-0.63	0.87	-0.48	-0.87	0.0096	1	0.25	0.32	0.64	0.66	-0.4	-0.63	-0.53	-0.22	-0.07	-0.33	-0.36	-0.34	-0.33	0.041
skew	-0.32	0.31	-0.26	-0.32	-0.21	0.25	1	0.98	-0.2	0.08	-0.43	-0.32	-0.17	-0.22	-0.081	-0.34	-0.062	-0.31	-0.3	-0.17
kurt	-0.32	0.35	-0.24	-0.35	-0.15	0.32	0.98	1	-0.13	0.11	-0.41	-0.32	-0.19	-0.2	-0.046	-0.3	-0.1	-0.27	-0.27	-0.21
sp.ent	-0.6	0.72	-0.5	-0.65	-0.17	0.64	-0.2	-0.13	1	0.87	-0.33	-0.6	-0.51	-0.31	-0.12	-0.29	-0.29	-0.32	-0.32	0.2
sfm	-0.78	0.84	-0.66	-0.77	-0.38	0.66	0.08	0.11	0.87	1	-0.49	-0.78	-0.42	-0.36	-0.19	-0.43	-0.29	-0.44	-0.43	0.21
mode	0.69	-0.53	0.68	0.59	0.49	-0.4	-0.43	-0.41	-0.33	-0.49	1	0.69	0.32	0.39	0.17	0.49	0.2	0.48	0.47	-0.18
centroid	1	-0.74	0.93	0.91	0.74	-0.63	-0.32	-0.32	-0.6	-0.78	0.69	1	0.46	0.38	0.27	0.54	0.23	0.52	0.52	-0.22
meanfun	0.46	-0.47	0.41	0.55	0.16	-0.53	-0.17	-0.19	-0.51	-0.42	0.32	0.46	1	0.34	0.31	0.27	0.16	0.28	0.28	-0.055
minfun	0.38	-0.35	0.34	0.32	0.26	-0.22	-0.22	-0.2	-0.31	-0.36	0.39	0.38	0.34	1	0.21	0.38	0.082	0.32	0.32	0.002
maxfun	0.27	-0.13	0.25	0.2	0.29	-0.07	-0.081	-0.046	-0.12	-0.19	0.17	0.27	0.31	0.21	1	0.34	-0.24	0.36	0.36	-0.36
meandom	0.54	-0.48	0.46	0.47	0.36	-0.33	-0.34	-0.3	-0.29	-0.43	0.49	0.54	0.27	0.38	0.34	1	0.1	0.81	0.81	-0.18
mindom	0.23	-0.36	0.19	0.3	-0.024	-0.36	-0.062	-0.1	-0.29	-0.29	0.2	0.23	0.16	0.082	-0.24	0.1	1	0.027	0.0087	0.2
maxdom	0.52	-0.48	0.44	0.46	0.34	-0.34	-0.31	-0.27	-0.32	-0.44	0.48	0.52	0.28	0.32	0.36	0.81	0.027	1	1	-0.43
dfrange	0.52	-0.48	0.44	0.45	0.34	-0.33	-0.3	-0.27	-0.32	-0.43	0.47	0.52	0.28	0.32	0.36	0.81	0.0087	1	1	-0.43
modindx	-0.22	0.12	-0.21	-0.14	-0.22	0.041	-0.17	-0.21	0.2	0.21	-0.18	-0.22	-0.055	0.002	-0.36	-0.18	0.2	-0.43	-0.43	1

```
fig = sns.FacetGrid(df, hue="label")
fig.map(sns.kdeplot, "meanfun", shade=True)
fig.add_legend()
```

```
<seaborn.axisgrid.FacetGrid at 0x7f62d72d6850>
```



```
data_train = df.sample(frac=0.7, random_state=1)
data_test = df.drop(data_train.index)
```

```
X_train = data_train.drop(['label'],axis=1)
y_train = data_train['label']
X_test = data_test.drop(['label'],axis=1)
y_test = data_test['label']
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
model = LogisticRegression(max_iter=1000)
model.fit(X_train,y_train)
prediction=model.predict(X_test)
print('The accuracy of the Logistic Regression is',accuracy_score(prediction,y_test))
```

```
     The accuracy of the Logistic Regression is 0.9084210526315789
```

Decision Tree Classifier

```
from sklearn.tree import DecisionTreeClassifier
model=DecisionTreeClassifier()
model.fit(X_train,y_train)
prediction=model.predict(X_test)
print('The accuracy of the Decision Tree is',accuracy_score(prediction,y_test))
```

```
     The accuracy of the Decision Tree is 0.9663157894736842
```

Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
classifier_rf = RandomForestClassifier(random_state=42, n_jobs=-1, max_depth=5, n_estimato
classifier_rf.fit(X_train, y_train)
print('The accuracy of the Random Forest is',classifier_rf.oob_score_)
```

```
     The accuracy of the Random Forest is 0.9729486023444545
```