In [1]:

```
import numpy as np # linear algebra
import matplotlib.pyplot as plt
import pandas as pd
```

In [2]:

```
df=pd.read_csv(r"C:\Users\Amarnath Gupta\Desktop\voice1.csv")
```

In [3]:

```
df.corr()
```

Out[3]:

	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt	sp.ent	s
meanfreq	1.000000	-0.739039	0.925445	0.911416	0.740997	-0.627605	-0.322327	-0.316036	-0.601203	-0.784
sd	-0.739039	1.000000	-0.562603	-0.846931	-0.161076	0.874660	0.314597	0.346241	0.716620	0.8380
median	0.925445	-0.562603	1.000000	0.774922	0.731849	-0.477352	-0.257407	-0.243382	-0.502005	-0.661
Q25	0.911416	-0.846931	0.774922	1.000000	0.477140	-0.874189	-0.319475	-0.350182	-0.648126	-0.766
Q75	0.740997	-0.161076	0.731849	0.477140	1.000000	0.009636	-0.206339	-0.148881	-0.174905	-0.378
IQR	-0.627605	0.874660	-0.477352	-0.874189	0.009636	1.000000	0.249497	0.316185	0.640813	0.6636
skew	-0.322327	0.314597	-0.257407	-0.319475	-0.206339	0.249497	1.000000	0.977020	-0.195459	0.0796
kurt	-0.316036	0.346241	-0.243382	-0.350182	-0.148881	0.316185	0.977020	1.000000	-0.127644	0.1098
sp.ent	-0.601203	0.716620	-0.502005	-0.648126	-0.174905	0.640813	-0.195459	-0.127644	1.000000	0.8664
sfm	-0.784332	0.838086	-0.661690	-0.766875	-0.378198	0.663601	0.079694	0.109884	0.866411	1.0000
mode	0.687715	-0.529150	0.677433	0.591277	0.486857	-0.403764	-0.434859	-0.406722	-0.325298	-0.485
centroid	1.000000	-0.739039	0.925445	0.911416	0.740997	-0.627605	-0.322327	-0.316036	-0.601203	-0.784
meanfun	0.460844	-0.466281	0.414909	0.545035	0.155091	-0.534462	-0.167668	-0.194560	-0.513194	-0.421
minfun	0.383937	-0.345609	0.337602	0.320994	0.258002	-0.222680	-0.216954	-0.203201	-0.305826	-0.362
maxfun	0.274004	-0.129662	0.251328	0.199841	0.285584	-0.069588	-0.080861	-0.045667	-0.120738	-0.192
meandom	0.536666	-0.482726	0.455943	0.467403	0.359181	-0.333362	-0.336848	-0.303234	-0.293562	-0.428
mindom	0.229261	-0.357667	0.191169	0.302255	-0.023750	-0.357037	-0.061608	-0.103313	-0.294869	-0.289
maxdom	0.519528	-0.482278	0.438919	0.459683	0.335114	-0.337877	-0.305651	-0.274500	-0.324253	-0.436
dfrange	0.515570	-0.475999	0.435621	0.454394	0.335648	-0.331563	-0.304640	-0.272729	-0.319054	-0.431
modindx	-0.216979	0.122660	-0.213298	-0.141377	-0.216475	0.041252	-0.169325	-0.205539	0.198074	0.2114

In [4]:

```
num_columns = df.shape[1]
x = df.iloc[:,:20].values
y = df.iloc[:,20].values
```

In [5]:

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
gender_labels = LabelEncoder()
y = gender_labels.fit_transform(y)
# lets see which is 0 and which is 1
print(list(gender_labels.inverse_transform([0,1])))
```

```
['female', 'male']
```

In [6]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2, random_state = 0)
```

In [7]:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

In [17]:

```
import keras
from keras.models import Sequential
from keras.layers import Dense

classifier = Sequential()
classifier.add(Dense(units = 1000, activation = 'relu', kernel_initializer = 'uniform', input_shape
= (20,)))
classifier.add(Dense(units = 11, activation = 'relu', kernel_initializer = 'uniform'))
classifier.add(Dense(units = 1, activation = 'sigmoid', kernel_initializer = 'uniform', input_shape
= (20,)))
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
classifier.fit(x_train, y_train, batch_size =8, epochs =8)
```

```
Epoch 1/8
2534/2534 [==============================] - 2s 626us/step - loss: 0.1890 - acc: 0.9459
Epoch 2/8
2534/2534 [==============================] - 1s 256us/step - loss: 0.0781 - acc: 0.9755
Epoch 3/8
2534/2534 [==============================] - 1s 257us/step - loss: 0.0678 - acc: 0.9767
Epoch 4/8
2534/2534 [==============================] - 1s 253us/step - loss: 0.0653 - acc: 0.9791
Epoch 5/8
2534/2534 [==============================] - 1s 256us/step - loss: 0.0622 - acc: 0.9787
Epoch 6/8
2534/2534 [==============================] - 1s 279us/step - loss: 0.0574 - acc: 0.9803
Epoch 7/8
2534/2534 [==============================] - 1s 291us/step - loss: 0.0568 - acc: 0.9799
Epoch 8/8
2534/2534 [==============================] - 1s 282us/step - loss: 0.0526 - acc: 0.9822
```

Out[17]:

```
<keras.callbacks.History at 0x1814f0c5cc0>
```

In [16]:

```
y_pred = classifier.predict(x_test)
y_pred = (y_pred > 0.5)
```

In [10]:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[298   3]
 [  8 325]]
```

In [16]:

```
import matplotlib.pyplot as plt
plt.matshow(cm)
plt.colorbar()
```

Out[16]:

```
<matplotlib.colorbar.Colorbar at 0x1a8449e2080>
```