CS7602 - MACHINE LEARNING

ASSIGNMENT 2

SUBMITTED BY

JAYASREE LAKSHMI NARAYAN 2016103033

AKHILA G P 2016103503

DATE : 28-03-2019

CONTENTS

DATASET USED

1. GENDER CLASSIFICATION BASED ON VOICE

A DESCRIPTION ON THE DATASET UNDER STUDY

GENDER DETECTION BASED ON VOICE

DESCRIPTION AND BASIC IDEA

In [1]:
```
import pandas as pd
df = pd.read_csv('voice.csv')
print(df.head(5))
```

```
   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   274.402906  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  label
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]
```

In [7]:
```
print "SHAPE : ",
print df.shape
print "NULL VALUES : ",
print df.isnull().values.any()
```

```
SHAPE :  (3168, 21)
NULL VALUES :  False
```

```
In [3]: print(df.describe())

          meanfreq           sd       median          Q25          Q75  \
count  3168.000000  3168.000000  3168.000000  3168.000000  3168.000000
mean      0.180907     0.057126     0.185621     0.140456     0.224765
std       0.029918     0.016652     0.036360     0.048680     0.023639
min       0.039363     0.018363     0.010975     0.000229     0.042946
25%       0.163662     0.041954     0.169593     0.111087     0.208747
50%       0.184838     0.059155     0.190032     0.140286     0.225684
75%       0.199146     0.067020     0.210618     0.175939     0.243660
max       0.251124     0.115273     0.261224     0.247347     0.273469

               IQR         skew         kurt       sp.ent          sfm  \
count  3168.000000  3168.000000  3168.000000  3168.000000  3168.000000
mean      0.084309     3.140168    36.568461     0.895127     0.408216
std       0.042783     4.240529   134.928661     0.044980     0.177521
min       0.014558     0.141735     2.068455     0.738651     0.036876
25%       0.042560     1.649569     5.669547     0.861811     0.258041
50%       0.094280     2.197101     8.318463     0.901767     0.396335
75%       0.114175     2.931694    13.648905     0.928713     0.533676
max       0.252225    34.725453  1309.612887     0.981997     0.842936

              mode     centroid      meanfun       minfun       maxfun  \
count  3168.000000  3168.000000  3168.000000  3168.000000  3168.000000
mean      0.165282     0.180907     0.142807     0.036802     0.258842
std       0.077203     0.029918     0.032304     0.019220     0.030077
min       0.000000     0.039363     0.055565     0.009775     0.103093
25%       0.118016     0.163662     0.116998     0.018223     0.253968
50%       0.186599     0.184838     0.140519     0.046110     0.271186
75%       0.221104     0.199146     0.169581     0.047904     0.277457
max       0.280000     0.251124     0.237636     0.204082     0.279114

           meandom       mindom       maxdom      dfrange      modindx
count  3168.000000  3168.000000  3168.000000  3168.000000  3168.000000
mean      0.829211     0.052647     5.047277     4.994630     0.173752
std       0.525205     0.063299     3.521157     3.520039     0.119454
min       0.007812     0.004883     0.007812     0.000000     0.000000
25%       0.419828     0.007812     2.070312     2.044922     0.099766
50%       0.765795     0.023438     4.992188     4.945312     0.139357
75%       1.177166     0.070312     7.007812     6.992188     0.209183
max       2.957682     0.458984    21.867188    21.843750     0.932374
```

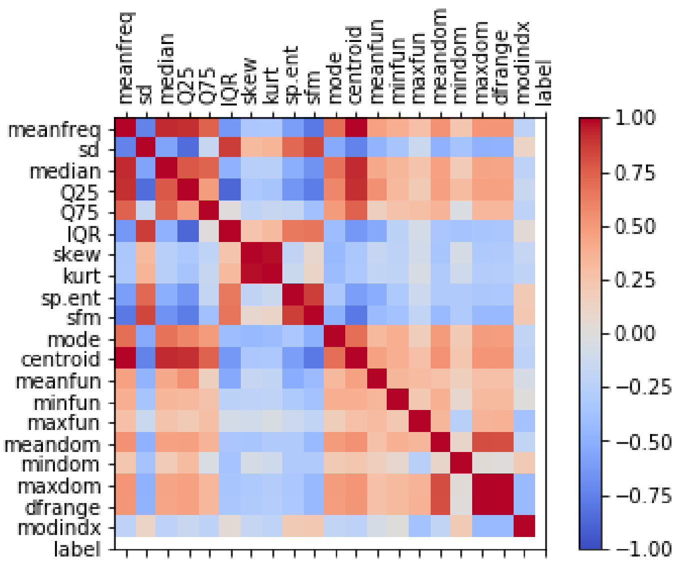


Illustration 1: Correlation matrix

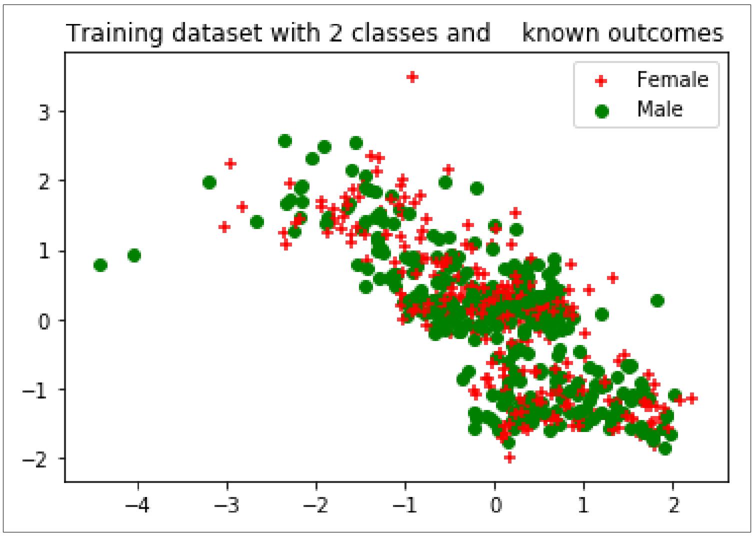


Illustration 2: Original data distribution

The jupyter notebook with the code is uploaded in Github and the link for the document is https://github.com/Akhilagp/ML_Assignment.

PROCEDURE:

- Support Vectors are the co-ordinates of individual observation. SVM is a frontier which best segregates the two classes (hyper-plane/ line).
- The hyper-plane is selected in such a way that it segregates the two classes better.
- When all the hyper-planes fit well, the best one for classification is chosen by maximizing the distance between nearest data points.
- Kernel trick is used when non linear hyper-planes are needed.
- PARAMETERS VARIED For Understanding
    1. Various Kernels
    2. Gamma value (tuning)
    3. The ‘C’ parameter – soft margin cost function (tuning)
    4. Degree of polynomial kernel.
    5. The number of Principal Components

OUTPUT:

Classifer	Accuracy with default parameters	Hyper-parameters tuning	
		C	**Accuracy**
SVC with Linear kernel	0.9694	0.1	0.9700
		0.2	0.9691
		0.3	0.9690
		0.4	0.9690
		0.5	0.9694
SVC with RBF kernel	0.9659	**Gamma**	**Accuracy**
		0.01	0.96815
		0.02	0.9678
		0.03	0.9678
		0.04	0.9668
		0.05	0.9659
SVC with polynomial kernel	0.9457	**Degree**	**Accuracy**
		2	0.8506
		3	0.9457
		4	0.8312
		5	0.8659
		6	0.7747

INFERENCE:

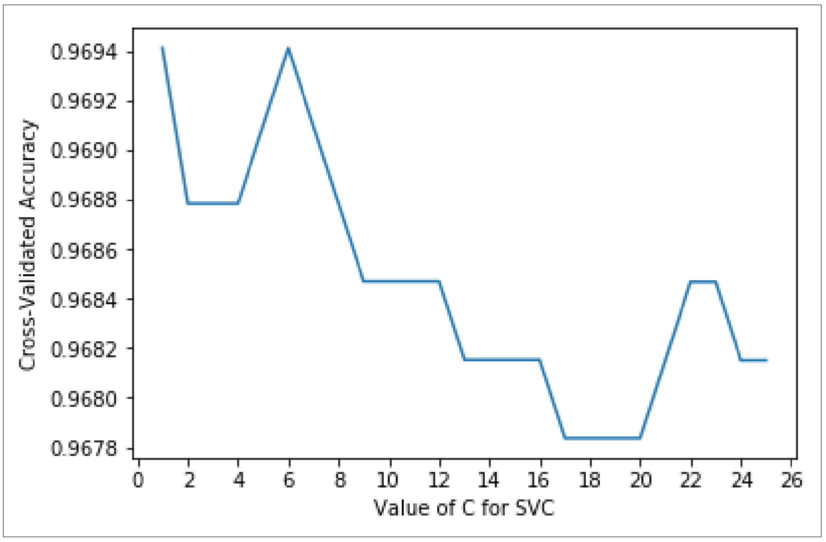


Illustration 3: Tuning soft-cost parameter

From illustration 1, it is evident that around C=0-1 and C=7-8, the accuracy hits the peak and at C=2 it falls. Varying the the values for C between 0 and 1, the exact value of C where the accuracy is the highest is found (C=0.1)

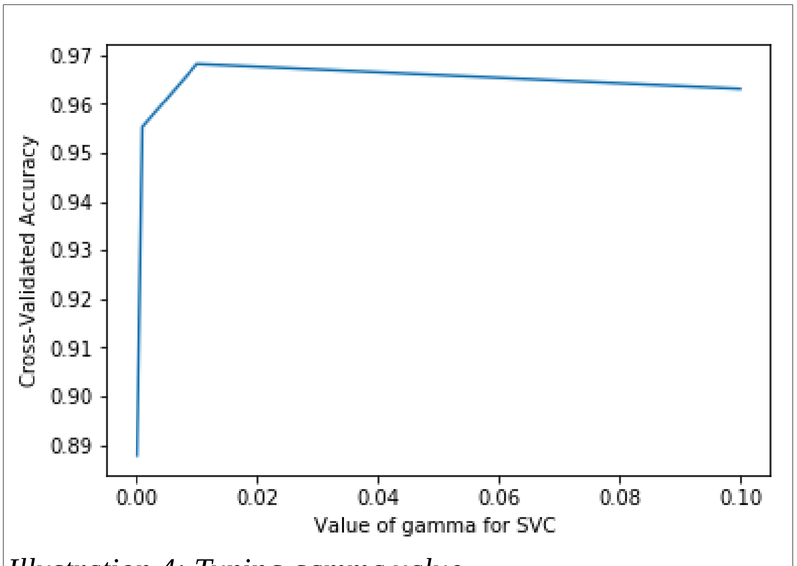


Illustration 4: Tuning gamma value

From illustration 2, around gamma = 0.0 and 0.02, the accuracy increases and starts to fall after 0.02. In turn, tuning the parameter with values between 0 and 0.02 the highest accuracy is obtained at gamma = 0.01.

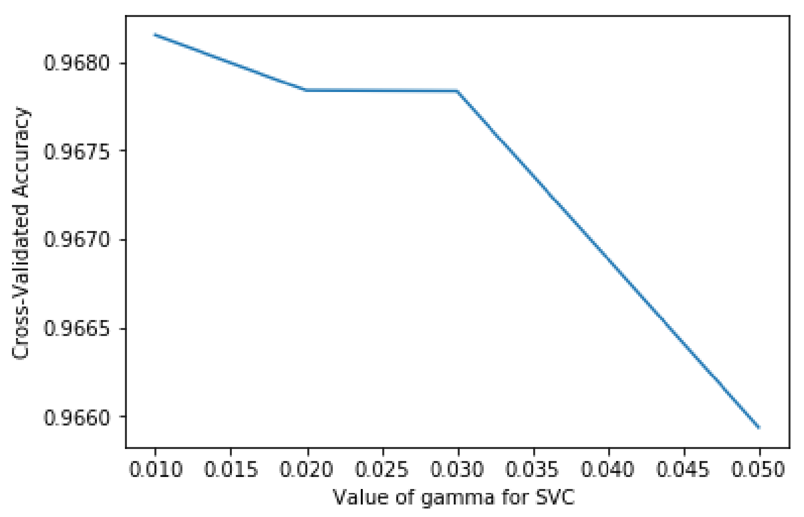


Illustration 5: Finest tuning

From illustration 4, it is evident that the 3rd degree polynomial (default value) gives the maximum accuracy.

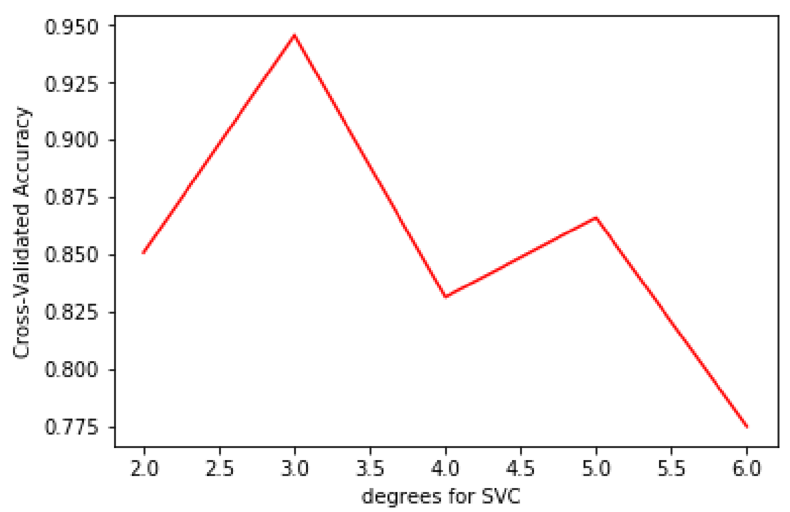


Illustration 6: Degree tuning

PRINICIPAL COMPONENT ANALYSIS

ALGORITHM

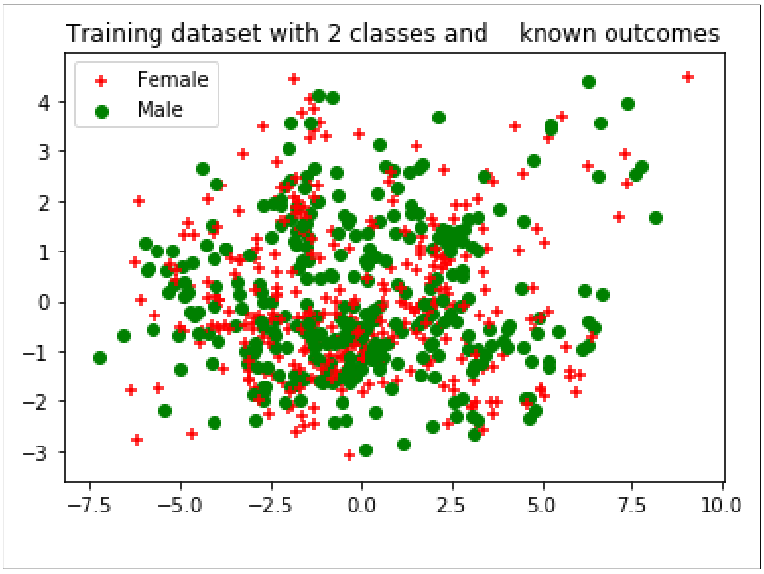
The Principal Components Analysis Algorithm

- Write N datapoints $\mathbf{x}_i = (\mathbf{x}_{1i}, \mathbf{x}_{2i}, \ldots, \mathbf{x}_{Mi})$ as row vectors
- Put these vectors into a matrix $\mathbf{X}$ (which will have size $N \times M$)
- Centre the data by subtracting off the mean of each column, putting it into matrix $\mathbf{B}$
- Compute the covariance matrix $\mathbf{C} = \frac{1}{N}\mathbf{B}^T\mathbf{B}$
- Compute the eigenvalues and eigenvectors of $\mathbf{C}$, so $\mathbf{V}^{-1}\mathbf{C}\mathbf{V} = \mathbf{D}$, where $\mathbf{V}$ holds the eigenvectors of $\mathbf{C}$ and $\mathbf{D}$ is the $M \times M$ diagonal eigenvalue matrix
- Sort the columns of $\mathbf{D}$ into order of decreasing eigenvalues, and apply the same order to the columns of V
- Reject those with eigenvalue less than some η, leaving L dimensions in the data

OUTPUT

Classifer	**PCA**	
	No. of Principal Components	**Accuracy**
SVC with Linear kernel	6	0.88328
	8	0.9526
	10	0.9763
	12	0.9842
	14	0.9668
SVC with RBF kernel	6	0.8880
	8	0.9258
	10	0.9495
	12	0.9637

	14	0.9558
SVC with polynomial kernel	6	0.9274
	8	0.9479
	10	0.9558
	12	0.9495
	14	0.9463



INFERENCE:

- For various kernels, the number of principal components are varied, and the highest accuracy is found.
- For Linear SVM, when using PCA and reducing the dimensionality, the accuracy has increased by 1% (Number of PC = 12).
- For polynomial SVM, when using PCA and reducing the dimensionality, the accuracy has increased by 2% (Number of PC = 10).