```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files spambase.data
• spambase.data(n/a) - 702942 bytes, last modified: 3/22/2021 - 100% done
Saving spambase.data to spambase.data
```

```
import numpy as np
data = np.genfromtxt("spambase.data", delimiter=",", skip_header=0)
data
```

```
array([[0.000e+00, 6.400e-01, 6.400e-01, ..., 6.100e+01, 2.780e+02,
        1.000e+00],
       [2.100e-01, 2.800e-01, 5.000e-01, ..., 1.010e+02, 1.028e+03,
        1.000e+00],
       [6.000e-02, 0.000e+00, 7.100e-01, ..., 4.850e+02, 2.259e+03,
        1.000e+00],
       ...,
       [3.000e-01, 0.000e+00, 3.000e-01, ..., 6.000e+00, 1.180e+02,
        0.000e+00],
       [9.600e-01, 0.000e+00, 0.000e+00, ..., 5.000e+00, 7.800e+01,
        0.000e+00],
       [0.000e+00, 0.000e+00, 6.500e-01, ..., 5.000e+00, 4.000e+01,
        0.000e+00]])
```

```
feature=data[0:,0:-1]
feature
```

```
array([[0.000e+00, 6.400e-01, 6.400e-01, ..., 3.756e+00, 6.100e+01,
        2.780e+02],
       [2.100e-01, 2.800e-01, 5.000e-01, ..., 5.114e+00, 1.010e+02,
        1.028e+03],
       [6.000e-02, 0.000e+00, 7.100e-01, ..., 9.821e+00, 4.850e+02,
        2.259e+03],
       ...,
       [3.000e-01, 0.000e+00, 3.000e-01, ..., 1.404e+00, 6.000e+00,
        1.180e+02],
       [9.600e-01, 0.000e+00, 0.000e+00, ..., 1.147e+00, 5.000e+00,
        7.800e+01],
       [0.000e+00, 0.000e+00, 6.500e-01, ..., 1.250e+00, 5.000e+00,
        4.000e+01]])
```

```
result=data[0:,-1:]
resultdim=result.reshape(-1)
print(resultdim.shape)
print(resultdim)
```

```
(4601,)
[1. 1. 1. ... 0. 0. 0.]
```

```
from sklearn.model_selection import train_test_split

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(feature, resultdim, test_size=0.3,rand
```

```
print(X_train)
print(X_train.shape)
print(y_train.shape)
print(y_train)
```

```
[[  0.      0.      0.    ...   3.857  28.    162.   ]
 [  0.      0.8     0.    ...   3.115  19.     81.   ]
 [  0.      0.      0.    ...   1.625   3.     13.   ]
 ...
 [  0.      0.      0.    ...   3.272  23.     36.   ]
 [  0.      0.      0.    ...   3.     18.     72.   ]
 [  0.      0.      0.    ...   1.625   9.     26.   ]]
(3220, 57)
(3220,)
[0. 1. 0. ... 1. 0. 0.]
```

```
from sklearn import svm
from sklearn import metrics
aray=[0.01,0.10,1.00,10.00,100.00,1000.00]
#Create a svm Classifier
for i in aray:
  clf = svm.SVC(C=i,kernel='linear') # Linear Kernel
  #Train the model using the training sets
  clf.fit(X_train, y_train)
  y_pred = clf.predict(X_test)
  print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.9102099927588704
Accuracy: 0.9203475742215785
Accuracy: 0.9268645908761767
Accuracy: 0.9268645908761767
Accuracy: 0.9210716871832005
Accuracy: 0.9210716871832005
```

```
aray=[0.01,0.10,1.00,10.00,100.00,1000.00,10000.00,100000.00,1000000.00,10000000.00]
for i in aray:
  clf = svm.SVC(C=i,kernel='rbf') # Linear Kernel
  #Train the model using the training sets
  clf.fit(X_train, y_train)
  y_pred = clf.predict(X_test)
  print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.6705286024619841
Accuracy: 0.6951484431571325
Accuracy: 0.719044170890659
Accuracy: 0.7299058653149891
Accuracy: 0.8124547429398986
Accuracy: 0.9058653149891384
Accuracy: 0.9290369297610427
Accuracy: 0.9326574945691528
Accuracy: 0.9326574945691528
Accuracy: 0.9312092686459088
```

```
#from sklearn import svm
```

```
#from sklearn import metrics
aray=[0.01,0.10,1.00,10.00,100.00,1000.00,10000.00,100000.00,1000000.00,10000000.00,100000
#Create a svm Classifier
for i in aray:
  clf = svm.SVC(C=i,kernel='poly',degree=2) # Linear Kernel
  #Train the model using the training sets
  clf.fit(X_train, y_train)
  y_pred = clf.predict(X_test)
  print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.6393917451122375
Accuracy: 0.6567704561911658
Accuracy: 0.66545981173063
Accuracy: 0.6683562635771181
Accuracy: 0.7139753801593048
Accuracy: 0.7834902244750181
Accuracy: 0.837074583635047
Accuracy: 0.887762490948588
Accuracy: 0.9174511223750905
Accuracy: 0.9232440260680667
```

Executing (42m 32s) Cell