Data Import and Cleaning

In [1]:

```
import pandas as pd
import numpy as np
import sklearn
from sklearn.model_selection import KFold
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import MinMaxScaler
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import LinearSVC
import matplotlib.pyplot as plt
from sklearn import metrics
from sklearn.ensemble import AdaBoostClassifier
scaler = MinMaxScaler()

import matplotlib.pyplot as plt
from matplotlib import cm
from math import log10

from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

from sklearn.metrics import roc_auc_score

from sklearn import metrics
from sklearn.metrics import roc_curve

from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import train_test_split
```

In [2]:

```
counter_strike_data = pd.read_csv("CSGOComplete.csv")
```

In [3]:

```
#Data Cleaning
counter_strike_data  = counter_strike_data.drop(['Day','M
onth','Year','Date'] , axis = 1)
x_counter_strike_data = counter_strike_data.drop(labels =
['Result'],axis = 1)
y_counter_strike_data = counter_strike_data[['Result']]
```

In [4]:

```
y_counter_strike_data = y_counter_strike_data['Result']
len_yResult = len(y_counter_strike_data)
for i in range(0, len_yResult):
    if(y_counter_strike_data[i] == 'Win'):
        y_counter_strike_data[i] = '1'
    elif(y_counter_strike_data[i] == 'Lost'):
        y_counter_strike_data[i] = '0'
    elif(y_counter_strike_data[i]== 'Tie'):
        y_counter_strike_data[i] = '2'
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\IPython\c
ore\interactiveshell.py:3296: SettingWithCop
yWarning:
A value is trying to be set on a copy of a s
lice from a DataFrame

See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy
  exec(code_obj, self.user_global_ns, self.u
ser_ns)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\ipykernel
_launcher.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a s
lice from a DataFrame

See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy
  import sys
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\ipykernel
_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a s
lice from a DataFrame

See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy
  """
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\ipykernel
_launcher.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a s
lice from a DataFrame
```

```
See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy
  if __name__ == '__main__':
```

In [5]:

```
x_counter_strike_data_copy = x_counter_strike_data
y_counter_strike_data_copy = y_counter_strike_data
```

In [6]:

```
colNames = x_counter_strike_data.columns

for i in range(0, len(colNames)):
    if(colNames[i]!="Map"):
        x_counter_strike_data[colNames[i]] = scaler.fit_t
ransform(x_counter_strike_data[colNames[i]]

.values.reshape(-1,1))
```

In [7]:

```
mapColumn = x_counter_strike_data['Map']
mapColumnLen = len(x_counter_strike_data['Map'])

for i in range(0,mapColumnLen):
    if(mapColumn[i] == 'Mirage'):
        mapColumn[i] = 0
    elif(mapColumn[i] == 'Dust II'):
        mapColumn[i] = 1
    elif(mapColumn[i] == 'Cache'):
        mapColumn[i] = 2
    elif(mapColumn[i] == 'Overpass'):
        mapColumn[i] = 3
    elif(mapColumn[i] == 'Cobblestone'):
        mapColumn[i] = 4
    elif(mapColumn[i] == 'Inferno'):
        mapColumn[i] = 5
    elif(mapColumn[i] == 'Austria'):
        mapColumn[i] = 6
    elif(mapColumn[i] == 'Canals'):
        mapColumn[i] = 7
    elif(mapColumn[i] == 'Nuke'):
        mapColumn[i] = 8
    elif(mapColumn[i] == 'Italy'):
        mapColumn[i] = 9
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\ipykernel
_launcher.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a s
lice from a DataFrame

See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy

c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\ipykernel
_launcher.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a s
lice from a DataFrame

See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy

c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\ipykernel
_launcher.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a s
lice from a DataFrame

See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy
  # Remove the CWD from sys.path while we lo
ad stuff.
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\ipykernel
_launcher.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a s
lice from a DataFrame
```

```
See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy
  if sys.path[0] == '':
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\ipykernel
_launcher.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a s
lice from a DataFrame

See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy

c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\ipykernel
_launcher.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a s
lice from a DataFrame

See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy
  app.launch_new_instance()
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\ipykernel
_launcher.py:18: SettingWithCopyWarning:
A value is trying to be set on a copy of a s
lice from a DataFrame

See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\ipykernel
_launcher.py:20: SettingWithCopyWarning:
A value is trying to be set on a copy of a s
```

```
lice from a DataFrame

See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\ipykernel
_launcher.py:22: SettingWithCopyWarning:
A value is trying to be set on a copy of a s
lice from a DataFrame

See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\ipykernel
_launcher.py:24: SettingWithCopyWarning:
A value is trying to be set on a copy of a s
lice from a DataFrame

See the caveats in the documentation: htt
p://pandas.pydata.org/pandas-docs/stable/ind
exing.html#indexing-view-versus-copy
```

In [8]:

```
#Converting to array
#x_counter_strike_data = np.array(x_counter_strike_data)
#y_counter_strike_data = np.array(y_counter_strike_data)

#kf = KFold(n_splits=3, random_state=50, shuffle=False)
#kf.get_n_splits(x_counter_strike_data)
#print(kf)

#for train_index, test_index in kf.split(x_counter_strike
_data):
 #  print("TRAIN:", train_index, "TEST:", test_index)
  # X_train, X_test = x_counter_strike_data[train_index],
x_counter_strike_data[test_index]
   #y_train, y_test = y_counter_strike_data[train_index],
y_counter_strike_data[test_index]


X_train, X_test, y_train, y_test = train_test_split(x_cou
nter_strike_data, y_counter_strike_data, test_size=0.3,
                                                    random
_state=1)
```

SVM - Counter strike - Training data

In [9]:

```
from sklearn.svm import SVC
number_of_iter = [10,100,1000,5000,10000,15000,20000]
kernelList = ['linear','rbf','poly']
tolerance = [0.001,0.01,0.1,1]
accuracyList = []

kernelList_final = []
tolerance_final = []
iterationList_final = []

my_step = 0
for i in range(0,len(kernelList)):
    for j in range(0,len(tolerance)):
        for k in range(0,len(number_of_iter)):
            linear_fit = SVC(gamma='scale',random_state=5
0, max_iter=number_of_iter[k],tol=tolerance[j],
                                                      ker
nel = kernelList[i])

            linear_fit.fit(X_train, y_train)
            predicted_svm = linear_fit.predict(X_train)
            cm = confusion_matrix(y_train, predicted_svm)

            kernelList_final.append(kernelList[i])
            tolerance_final.append(tolerance[j])
            iterationList_final.append(number_of_iter[k])
            accuracyList.append((cm[0][0] + cm[1][1]) / n
p.sum(cm))

            my_step = my_step + 1
            print("done:",my_step,"/",len(kernelList) * l
en(tolerance) * len(number_of_iter))

print('Kernel Tolerance Iterations Accuracy')
for l in range(0,len(kernelList) * len(tolerance) * len(n
umber_of_iter)):
    print(kernelList_final[l],tolerance_final[l],iterationList_final[l],accuracyList[l])
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)

done: 1 / 84
done: 2 / 84
done: 3 / 84
done: 4 / 84
done: 5 / 84
done: 6 / 84
done: 7 / 84
done: 8 / 84
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

```
done: 9 / 84
done: 10 / 84
done: 11 / 84
done: 12 / 84
done: 13 / 84
done: 14 / 84
done: 15 / 84
done: 16 / 84
done: 17 / 84


c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

```
done: 18 / 84
done: 19 / 84
done: 20 / 84
done: 21 / 84
done: 22 / 84
done: 23 / 84
done: 24 / 84
done: 25 / 84
done: 26 / 84
done: 27 / 84
done: 28 / 84
done: 29 / 84
done: 30 / 84
done: 31 / 84
done: 32 / 84
done: 33 /

c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

```
84
done: 34 / 84
done: 35 / 84
done: 36 / 84
done: 37 / 84
done: 38 / 84
done:

c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)

39 / 84
done: 40 / 84
done: 41 / 84
done: 42 / 84
done: 43 / 84
done: 44 / 84
done: 45 / 84
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)


done: 46 / 84
done: 47 / 84
done: 48 / 84
done: 49 / 84
done: 50 / 84
done: 51 / 84

c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

```
done: 52 / 84
done: 53 / 84
done: 54 / 84
done: 55 / 84
done: 56 / 84
done:

c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=5000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

```
 57 / 84
done: 58 / 84
done: 59 / 84
done: 60 / 84
done: 61 / 84
done: 62 / 84
done: 63 / 84
done: 64 / 84
done: 65 / 84
done: 66 / 84
done: 67 / 84
done: 68 / 84

c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

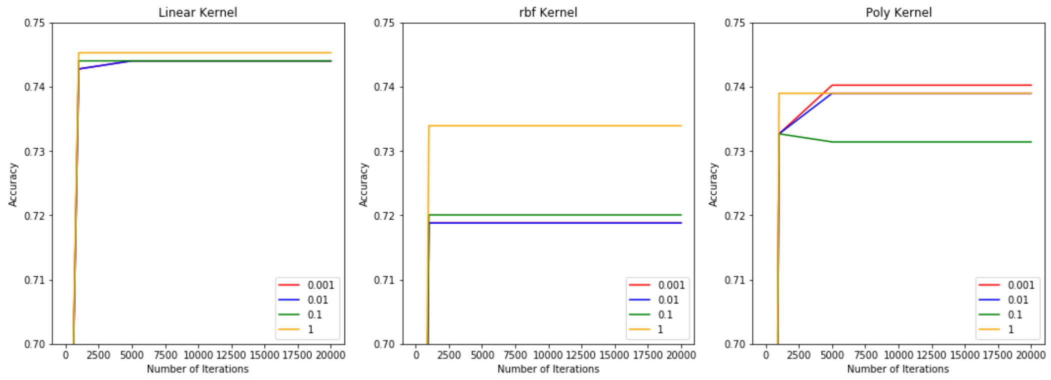
```
done: 69 / 84
done: 70 / 84
done: 71 / 84
done: 72 / 84
done: 73 / 84
done: 74 / 84
done: 75 / 84

c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

```
done: 76 / 84
done: 77 / 84
done: 78 / 84
done: 79 / 84
done: 80 / 84
done: 81 / 84
done: 82 / 84
done: 83 / 84


c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

```
done: 84 / 84
Kernel Tolerance Iterations Accuracy
linear 0.001 10 0.4098360655737705
linear 0.001 100 0.6443883984867591
linear 0.001 1000 0.742749054224464
linear 0.001 5000 0.7440100882723834
linear 0.001 10000 0.7440100882723834
linear 0.001 15000 0.7440100882723834
linear 0.001 20000 0.7440100882723834
linear 0.01 10 0.4098360655737705
linear 0.01 100 0.6443883984867591
linear 0.01 1000 0.742749054224464
linear 0.01 5000 0.7440100882723834
linear 0.01 10000 0.7440100882723834
linear 0.01 15000 0.7440100882723834
linear 0.01 20000 0.7440100882723834
linear 0.1 10 0.4098360655737705
linear 0.1 100 0.6443883984867591
linear 0.1 1000 0.7440100882723834
linear 0.1 5000 0.7440100882723834
linear 0.1 10000 0.7440100882723834
linear 0.1 15000 0.7440100882723834
linear 0.1 20000 0.7440100882723834
linear 1 10 0.4098360655737705
linear 1 100 0.6456494325346784
linear 1 1000 0.7452711223203027
linear 1 5000 0.7452711223203027
linear 1 10000 0.7452711223203027
linear 1 15000 0.7452711223203027
linear 1 20000 0.7452711223203027
rbf 0.001 10 0.4602774274905423
rbf 0.001 100 0.5031525851197982
rbf 0.001 1000 0.7187894073139974
rbf 0.001 5000 0.7187894073139974
rbf 0.001 10000 0.7187894073139974
rbf 0.001 15000 0.7187894073139974
rbf 0.001 20000 0.7187894073139974
rbf 0.01 10 0.4602774274905422З
rbf 0.01 100 0.5031525851197982
rbf 0.01 1000 0.7187894073139974
rbf 0.01 5000 0.7187894073139974
rbf 0.01 10000 0.7187894073139974
rbf 0.01 15000 0.7187894073139974
rbf 0.01 20000 0.7187894073139974
rbf 0.1 10 0.46027742749054223
rbf 0.1 100 0.5031525851197982
rbf 0.1 1000 0.7200504413619168
rbf 0.1 5000 0.7200504413619168
rbf 0.1 10000 0.7200504413619168
rbf 0.1 15000 0.7200504413619168
rbf 0.1 20000 0.7200504413619168
rbf 1 10 0.46027742749054223
rbf 1 100 0.5031525851197982
rbf 1 1000 0.733921815889029
rbf 1 5000 0.733921815889029
rbf 1 10000 0.733921815889029
rbf 1 15000 0.733921815889029
rbf 1 20000 0.733921815889029
poly 0.001 10 0.12610340479192939
poly 0.001 100 0.4426229508196721
poly 0.001 1000 0.7326607818411097
poly 0.001 5000 0.7402269861286255
poly 0.001 10000 0.7402269861286255
poly 0.001 15000 0.7402269861286255
poly 0.001 20000 0.7402269861286255
poly 0.01 10 0.12610340479192939
poly 0.01 100 0.4426229508196721
poly 0.01 1000 0.7326607818411097
poly 0.01 5000 0.7389659520807061
poly 0.01 10000 0.7389659520807061
poly 0.01 15000 0.7389659520807061
poly 0.01 20000 0.7389659520807061
poly 0.1 10 0.12610340479192939
poly 0.1 100 0.4426229508196721
poly 0.1 1000 0.7326607818411097
poly 0.1 5000 0.7313997477931904
poly 0.1 10000 0.7313997477931904
poly 0.1 15000 0.7313997477931904
poly 0.1 20000 0.7313997477931904
poly 1 10 0.12610340479192939
poly 1 100 0.4426229508196721
poly 1 1000 0.7389659520807061
poly 1 5000 0.7389659520807061
poly 1 10000 0.7389659520807061
poly 1 15000 0.7389659520807061
poly 1 20000 0.7389659520807061
```

In [10]:

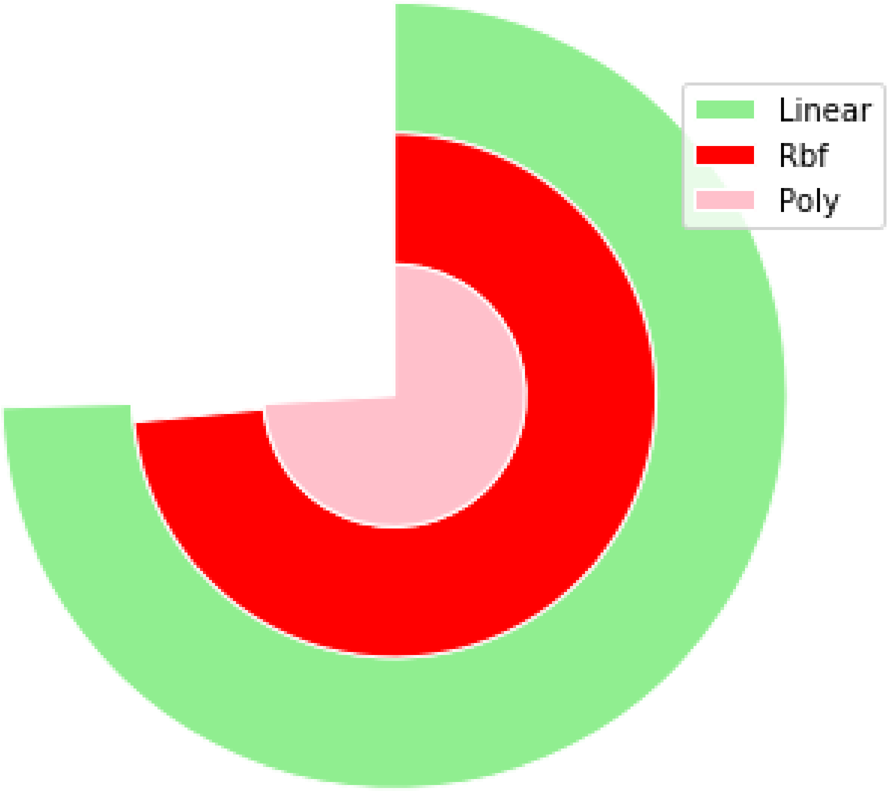
```
x_range = [10,100,1000,5000,10000,15000,20000]
plt.figure(figsize=(18,6))

#Linear kernel
plt.subplot(1, 3, 1)
plt.plot(x_range, accuracyList[0:7], color='r',label ='0.001')
plt.plot(x_range, accuracyList[7:14], color='b',label = '0.01')
plt.plot(x_range, accuracyList[14:21], color='g',label = '0.1')
plt.plot(x_range, accuracyList[21:28], color='orange',label ='1')
plt.ylim(0.70,0.75)

plt.title("Linear Kernel")
plt.legend(loc='lower right')
plt.xlabel('Number of Iterations')
plt.ylabel('Accuracy')


#rbf kernel
plt.subplot(1, 3, 2)
plt.plot(x_range, accuracyList[28:35], color='r',label = '0.001')
plt.plot(x_range, accuracyList[35:42], color='b',label = '0.01')
plt.plot(x_range, accuracyList[42:49], color='g',label = '0.1')
plt.plot(x_range, accuracyList[49:56], color='orange',label ='1')


plt.title("rbf Kernel")
plt.legend(loc='lower right')
plt.xlabel('Number of Iterations')
plt.ylabel('Accuracy')
plt.ylim(0.70,0.75)


#poly kernel
plt.subplot(1, 3, 3)
plt.plot(x_range, accuracyList[56:63], color='r',label = '0.001')
plt.plot(x_range, accuracyList[63:70], color='b',label = '0.01')
plt.plot(x_range, accuracyList[70:77], color='g',label = '0.1')
plt.plot(x_range, accuracyList[77:84], color='orange',label ='1')

plt.ylim(0.70,0.75)
plt.title("Poly Kernel")
plt.legend(loc='lower right')
plt.xlabel('Number of Iterations')
plt.ylabel('Accuracy')


plt.show()
```



In [11]:

```
print("Maximum accuracy in each kernel")
labels = ['Linear','Rbf','Poly']
data = [max(accuracyList[0:28]),max(accuracyList[28:56]),
max(accuracyList[56:84])]
#number of data points
n = len(data)
#find max value for full ring
k = 10 ** int(log10(max(data)))
m = k * (1 + max(data) // k)

#radius of donut chart
r = 1.5
#calculate width of each ring
w = r / n

#create colors along a chosen colormap
colors = ['lightgreen','red','pink']

#create figure, axis
fig, ax = plt.subplots()
ax.axis("equal")

#create rings of donut chart
for i in range(n):
    #hide labels in segments with textprops: alpha = 0 -
 transparent, alpha = 1 - visible
    innerring, _ = ax.pie([m - data[i], data[i]], radius
= r - i * w, startangle = 90, labels = ["", labels[i]], l
abeldistance = 1 - 1 / (1.5 * (n - i)), textprops = {"alp
ha": 0}, colors = ["white", colors[i]])
    plt.setp(innerring, width = w, edgecolor = "white")

plt.legend()
plt.show()
```

```
Maximum accuracy in each kernel
```



In [12]:

```
from sklearn.model_selection import cross_val_score

linear_fit = SVC(gamma='auto', kernel='linear', tol=0.01, max_iter=1000,
                            random_state=50)
linear_fit.fit(X_train, y_train)
predicted_svm = linear_fit.predict(X_test)
cm = confusion_matrix(y_test, predicted_svm)
cm
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

Out[12]:

```
array([[145,  32,   0],
       [ 36, 103,   0],
       [ 15,   9,   0]], dtype=int64)
```

In [13]:

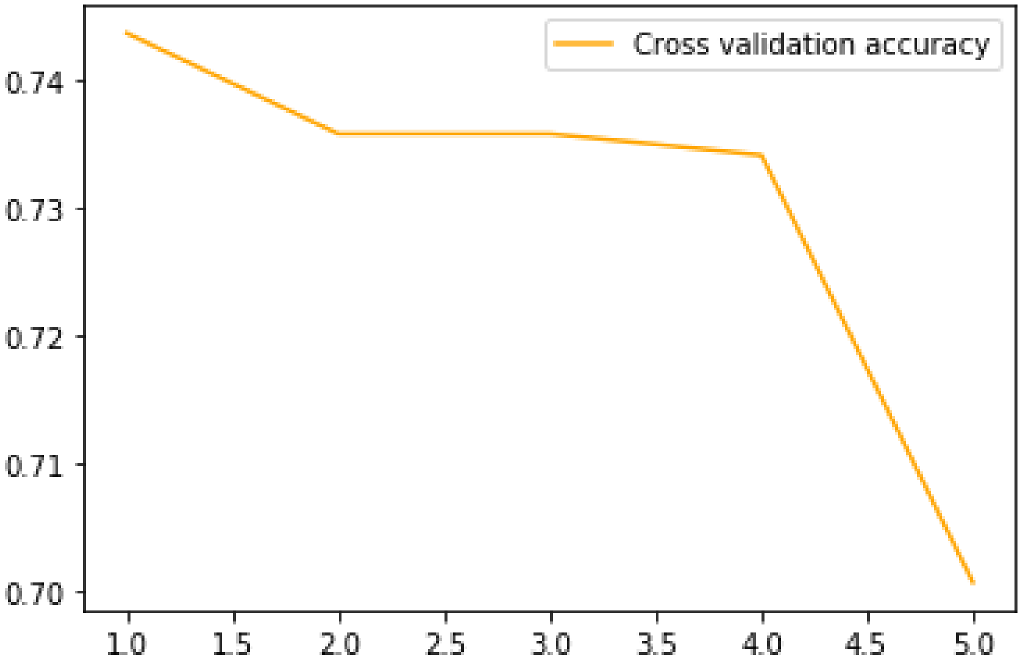
```
linear_fit = SVC(gamma='auto', kernel='linear', tol=0.01,
max_iter=1000,
                            random_state=50)
linear_fit.fit(X_train, y_train)

scores = cross_val_score(linear_fit, X_train, y_train, cv
=5)
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

In [14]:

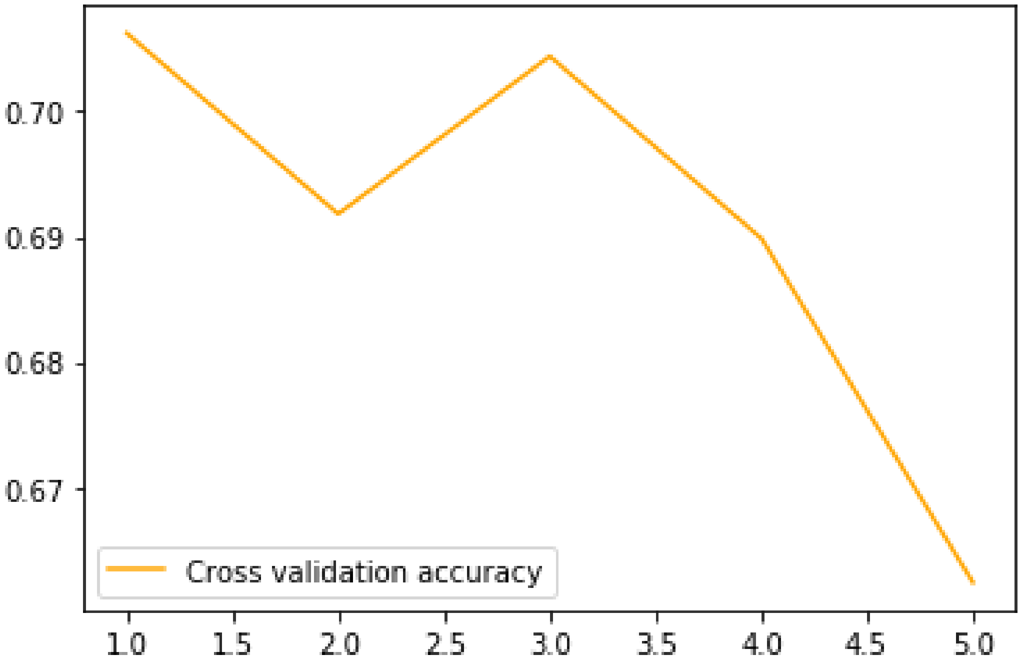
```
linear_fit = SVC(gamma='auto', kernel='linear', tol=0.01,
max_iter=1000,
                            random_state=50)
linear_fit.fit(X_train, y_train)
scores = cross_val_score(linear_fit, X_train, y_train, cv
=5)


x_range = list(range(1, 6))
plt.plot(x_range, scores, color='orange',label ='Cross va
lidation accuracy')
plt.legend()
plt.show()
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```



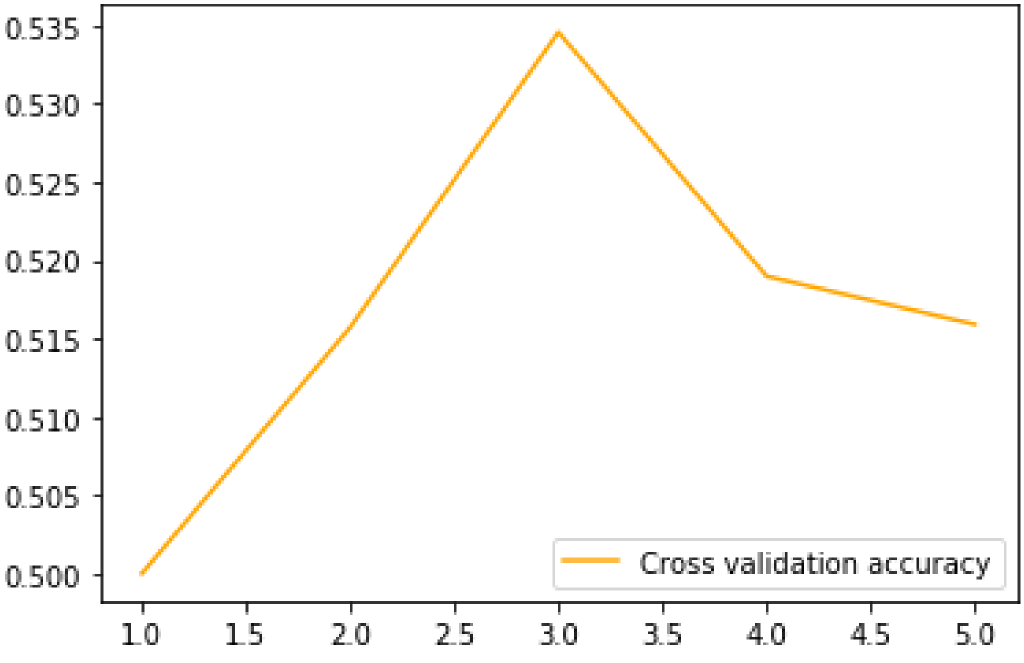
In [15]:

```
linear_fit = SVC(gamma='auto', kernel='rbf', tol=0.01, ma
x_iter=1000,
                            random_state=50)
linear_fit.fit(X_train, y_train)
scores = cross_val_score(linear_fit, X_train, y_train, cv
=5)


x_range = list(range(1, 6))
plt.plot(x_range, scores, color='orange',label ='Cross va
lidation accuracy')
plt.legend()
plt.show()
```



In [16]:

```
linear_fit = SVC(gamma='auto', kernel='poly', tol=0.01, max_iter=1000,
                            random_state=50)
linear_fit.fit(X_train, y_train)
scores = cross_val_score(linear_fit, X_train, y_train, cv=5)


x_range = list(range(1, 6))
plt.plot(x_range, scores, color='orange',label ='Cross validation accuracy')
plt.legend()
plt.show()
```



SVM - Counter strike - Validation data

In [17]:

```
number_of_iter = [10,100,1000,5000,10000,15000,20000]
kernelList = ['linear','rbf','poly']
tolerance = [0.001,0.01,0.1,1]
accuracyList = []

kernelList_final = []
tolerance_final = []
iterationList_final = []

my_step = 0
for i in range(0,len(kernelList)):
    for j in range(0,len(tolerance)):
        for k in range(0,len(number_of_iter)):
            linear_fit = SVC(gamma='scale',random_state=5
0, max_iter=number_of_iter[k],tol=tolerance[j],
                                                       ker
nel = kernelList[i])
            linear_fit.fit(X_train, y_train)
            predicted_svm = linear_fit.predict(X_test)
            cm = confusion_matrix(y_test, predicted_svm)

            kernelList_final.append(kernelList[i])
            tolerance_final.append(tolerance[j])
            iterationList_final.append(number_of_iter[k])
            accuracyList.append((cm[0][0] + cm[1][1]) / n
p.sum(cm))

            my_step = my_step + 1
            print("done:",my_step,"/",len(kernelList) * l
en(tolerance) * len(number_of_iter))

print('Kernel Tolerance Iterations Accuracy')
for l in range(0,len(kernelList) * len(tolerance) * len(n
umber_of_iter)):
    print(kernelList_final[l],tolerance_final[l],iterationList_final[l],accuracyList[l])
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
```

```
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

```
done: 1 / 84
done: 2 / 84
done: 3 / 84
done: 4 / 84
done: 5 / 84
done: 6 / 84
done: 7 / 84
done: 8 / 84
done: 9 / 84
done: 10 / 84
done: 11 / 84
done: 12 / 84
done: 13 / 84
done: 14 / 84
done: 15 / 84
done: 16 / 84
done: 17 / 84
done: 18 / 84
done: 19 / 84
done: 20 / 84
done: 21 / 84
done: 22 / 84
done: 23 / 84
done: 24 / 84
done: 25 / 84
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)

done: 26 / 84
done: 27 / 84
done: 28 / 84
done: 29 / 84
done: 30 / 84
done: 31 / 84
done:
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)


 32 / 84
done: 33 / 84
done: 34 / 84
done: 35 / 84
done: 36 / 84
done: 37 / 84
done: 38 / 84
done: 39 / 84
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)

done: 40 / 84
done: 41 / 84
done: 42 / 84
done: 43 / 84
done: 44 / 84
done: 45 / 84
done: 46 / 84
done: 47 / 84
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)


done: 48 / 84
done: 49 / 84
done: 50 / 84
done: 51 / 84
done: 52 / 84
done: 53 / 84
done: 54 / 84
done: 55 / 84
done: 56 / 84
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=5000).  Consider p
re-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
```

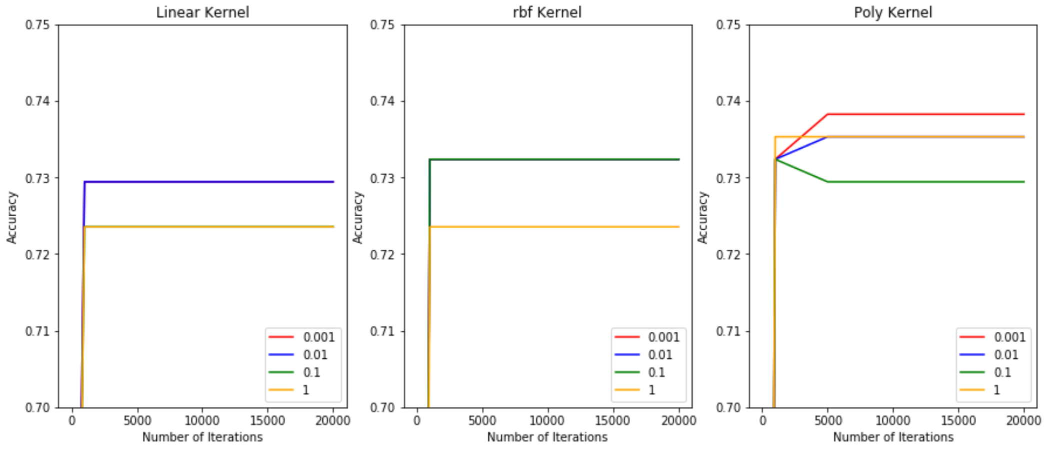
```
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler 
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)


done: 57 / 84
done: 58 / 84
done: 59 / 84
done: 60 / 84
done: 61 / 84
done: 62 / 84
done: 63 / 84
done: 64 / 84
done: 65 / 84
done: 66 / 84
done: 67 / 84
done: 68 / 84
done: 69 / 84
done: 70 / 84
done: 71 / 84
done: 72 / 84
done: 73 / 84
done: 74 / 84
```

```
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=1000).  Consider p
re-processing your data with StandardScaler 
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=10).  Consider pre
-processing your data with StandardScaler or
MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
c:\users\siddharth\appdata\local\programs\py
thon\python37-32\lib\site-packages\sklearn\s
vm\base.py:241: ConvergenceWarning: Solver t
erminated early (max_iter=100).  Consider pr
e-processing your data with StandardScaler o
r MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

```
done: 75 / 84
done: 76 / 84
done: 77 / 84
done: 78 / 84
done: 79 / 84
done: 80 / 84
done: 81 / 84
done: 82 / 84
done: 83 / 84
done: 84 / 84
Kernel Tolerance Iterations Accuracy
linear 0.001 10 0.43529411764705883
linear 0.001 100 0.611764705882353
linear 0.001 1000 0.7294117647058823
linear 0.001 5000 0.7294117647058823
linear 0.001 10000 0.7294117647058823
linear 0.001 15000 0.7294117647058823
linear 0.001 20000 0.7294117647058823
linear 0.01 10 0.43529411764705883
linear 0.01 100 0.611764705882353
linear 0.01 1000 0.7294117647058823
linear 0.01 5000 0.7294117647058823
linear 0.01 10000 0.7294117647058823
linear 0.01 15000 0.7294117647058823
linear 0.01 20000 0.7294117647058823
linear 0.1 10 0.43529411764705883
linear 0.1 100 0.611764705882353
linear 0.1 1000 0.7235294117647059
linear 0.1 5000 0.7235294117647059
linear 0.1 10000 0.7235294117647059
linear 0.1 15000 0.7235294117647059
linear 0.1 20000 0.7235294117647059
linear 1 10 0.43529411764705883
linear 1 100 0.6088235294117647
linear 1 1000 0.7235294117647059
linear 1 5000 0.7235294117647059
linear 1 10000 0.7235294117647059
linear 1 15000 0.7235294117647059
linear 1 20000 0.7235294117647059
rbf 0.001 10 0.4294117647058823
rbf 0.001 100 0.45588235294117646
rbf 0.001 1000 0.7323529411764705
rbf 0.001 5000 0.7323529411764705
rbf 0.001 10000 0.7323529411764705
rbf 0.001 15000 0.7323529411764705
rbf 0.001 20000 0.7323529411764705
rbf 0.01 10 0.4294117647058823
rbf 0.01 100 0.45588235294117646
rbf 0.01 1000 0.7323529411764705
rbf 0.01 5000 0.7323529411764705
rbf 0.01 10000 0.7323529411764705
rbf 0.01 15000 0.7323529411764705
rbf 0.01 20000 0.7323529411764705
rbf 0.1 10 0.4294117647058823
rbf 0.1 100 0.45588235294117646
rbf 0.1 1000 0.7323529411764705
rbf 0.1 5000 0.7323529411764705
rbf 0.1 10000 0.7323529411764705
rbf 0.1 15000 0.7323529411764705
rbf 0.1 20000 0.7323529411764705
rbf 1 10 0.4294117647058823
rbf 1 100 0.45588235294117646
rbf 1 1000 0.7235294117647059
rbf 1 5000 0.7235294117647059
rbf 1 10000 0.7235294117647059
rbf 1 15000 0.7235294117647059
rbf 1 20000 0.7235294117647059
poly 0.001 10 0.11470588235294117
poly 0.001 100 0.4088235294117647
poly 0.001 1000 0.7323529411764705
poly 0.001 5000 0.7382352941176471
poly 0.001 10000 0.7382352941176471
poly 0.001 15000 0.7382352941176471
poly 0.001 20000 0.7382352941176471
poly 0.01 10 0.11470588235294117
poly 0.01 100 0.4088235294117647
poly 0.01 1000 0.7323529411764705
poly 0.01 5000 0.7352941176470589
poly 0.01 10000 0.7352941176470589
poly 0.01 15000 0.7352941176470589
poly 0.01 20000 0.7352941176470589
poly 0.1 10 0.11470588235294117
poly 0.1 100 0.4088235294117647
poly 0.1 1000 0.7323529411764705
poly 0.1 5000 0.7294117647058823
poly 0.1 10000 0.7294117647058823
poly 0.1 15000 0.7294117647058823
poly 0.1 20000 0.7294117647058823
poly 1 10 0.11470588235294117
poly 1 100 0.4088235294117647
poly 1 1000 0.7352941176470589
poly 1 5000 0.7352941176470589
poly 1 10000 0.7352941176470589
poly 1 15000 0.7352941176470589
poly 1 20000 0.7352941176470589
```

In [18]:

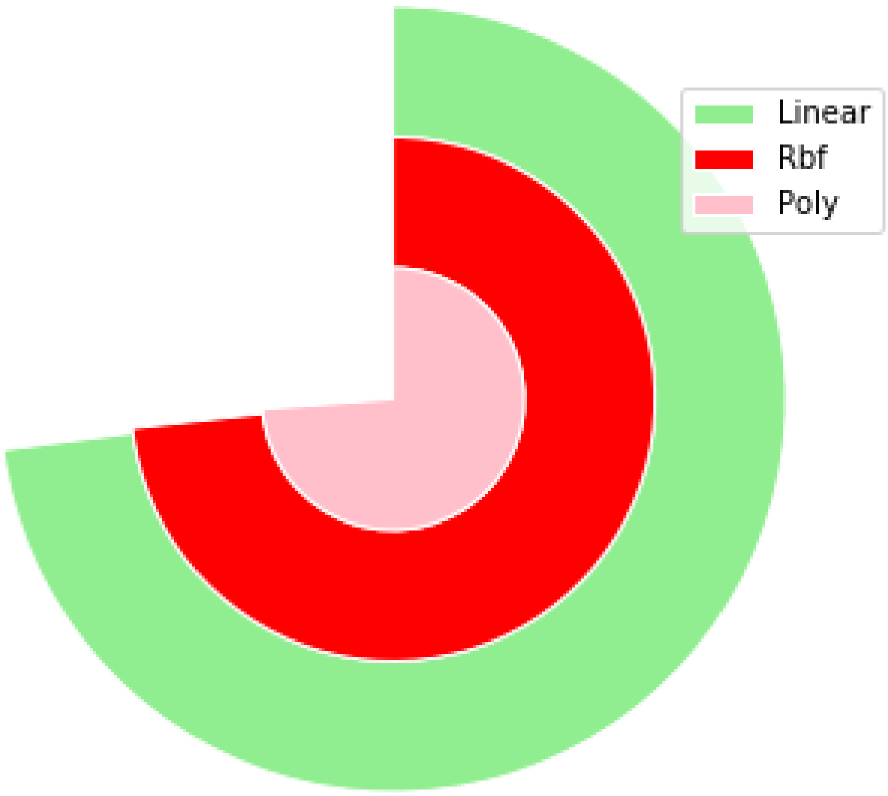
```
x_range = [10,100,1000,5000,10000,15000,20000]
plt.figure(figsize=(15,6))

#Linear kernel
plt.subplot(1, 3, 1)
plt.plot(x_range, accuracyList[0:7], color='r',label ='0.001')
plt.plot(x_range, accuracyList[7:14], color='b',label = '0.01')
plt.plot(x_range, accuracyList[14:21], color='g',label = '0.1')
plt.plot(x_range, accuracyList[21:28], color='orange',label ='1')

#plt.ylim(0.70,0.728)
plt.title("Linear Kernel")
plt.legend(loc='lower right')
plt.xlabel('Number of Iterations')
plt.ylabel('Accuracy')

plt.ylim(0.70,0.75)
#rbf kernel
plt.subplot(1, 3, 2)
plt.plot(x_range, accuracyList[28:35], color='r',label = '0.001')
plt.plot(x_range, accuracyList[35:42], color='b',label = '0.01')
plt.plot(x_range, accuracyList[42:49], color='g',label = '0.1')
plt.plot(x_range, accuracyList[49:56], color='orange',label ='1')

#plt.ylim(0.70,0.728)
plt.title("rbf Kernel")
plt.legend(loc='lower right')
plt.xlabel('Number of Iterations')
plt.ylabel('Accuracy')
plt.ylim(0.70,0.75)


#poly kernel
plt.subplot(1, 3, 3)
plt.plot(x_range, accuracyList[56:63], color='r',label =
'0.001')
plt.plot(x_range, accuracyList[63:70], color='b',label =
'0.01')
plt.plot(x_range, accuracyList[70:77], color='g',label =
'0.1')
plt.plot(x_range, accuracyList[77:84], color='orange',lab
el ='1')

plt.ylim(0.70,0.75)
plt.title("Poly Kernel")
plt.legend(loc='lower right')
plt.xlabel('Number of Iterations')
plt.ylabel('Accuracy')


plt.show()
```



In [19]:

```python
import matplotlib.pyplot as plt
from matplotlib import cm
from math import log10

print("Maximum accuracy in each kernel")
labels = ['Linear','Rbf','Poly']
data = [max(accuracyList[0:28]),max(accuracyList[28:56]),max(accuracyList[56:84])]
#number of data points
n = len(data)
#find max value for full ring
k = 10 ** int(log10(max(data)))
m = k * (1 + max(data) // k)

#radius of donut chart
r = 1.5
#calculate width of each ring
w = r / n

#create colors along a chosen colormap
colors = ['lightgreen','red','pink']

#create figure, axis
fig, ax = plt.subplots()
ax.axis("equal")

#create rings of donut chart
for i in range(n):
    #hide labels in segments with textprops: alpha = 0 - transparent, alpha = 1 - visible
    innerring, _ = ax.pie([m - data[i], data[i]], radius = r - i * w, startangle = 90, labels = ["", labels[i]], labeldistance = 1 - 1 / (1.5 * (n - i)), textprops = {"alpha": 0}, colors = ["white", colors[i]])
    plt.setp(innerring, width = w, edgecolor = "white")

plt.legend()
plt.show()
```

```
Maximum accuracy in each kernel
```



In [20]:

```
linear_fit = SVC(gamma='auto', kernel='linear', tol=0.001
, max_iter=5000,
                              random_state=50)
linear_fit.fit(X_train, y_train)
predicted_svm = linear_fit.predict(X_test)
cm = confusion_matrix(y_test, predicted_svm)
cm
```

Out[20]:

```
array([[145,  32,   0],
       [ 36, 103,   0],
       [ 15,   9,   0]], dtype=int64)
```

In [21]:

```
print("Accuracy", (cm[1][1] + cm[0][0]) / (cm[1][1] + cm[
0][0] + cm[0][1] + cm[1][0]) )
print("Sensitivity", cm[1][1] / (cm[1][1] + cm[1][0] ))
print("Specificity", cm[0][0] / (cm[0][0] + cm[0][1] ))
print("Precision", cm[1][1] / (cm[1][1] + cm[0][1] ))
```

```
Accuracy 0.7848101265822784
Sensitivity 0.7410071942446043
Specificity 0.8192090395480226
Precision 0.762962962962963
```

Decision tree - Counter Strike

In [22]:

```
training_depth_Accuracy = []
for i in range(0,10):
    clf = DecisionTreeClassifier(criterion="gini", max_depth=(i+1),random_state=50)
    clf.fit(X_train,y_train)
    y_pred_train = clf.predict(X_train)
    training_depth_Accuracy.append(metrics.accuracy_score(y_train, y_pred_train))

print(training_depth_Accuracy)
```

```
[0.6645649432534678, 0.691046658259773, 0.73
89659520807061, 0.7667087011349306, 0.800756
6204287516, 0.8852459016393442, 0.9155107187
894073, 0.935687263556116, 0.967213114754098
3, 0.982345523291298]
```

In [23]:

```
validation_depth_Accuracy = []
for i in range(0,10):
    clf = DecisionTreeClassifier(criterion="gini", max_de
pth=(i+1),random_state=50)
    clf.fit(X_train,y_train)
    y_pred_test = clf.predict(X_test)
    validation_depth_Accuracy.append(metrics.accuracy_sco
re(y_test, y_pred_test))

print(validation_depth_Accuracy)
```

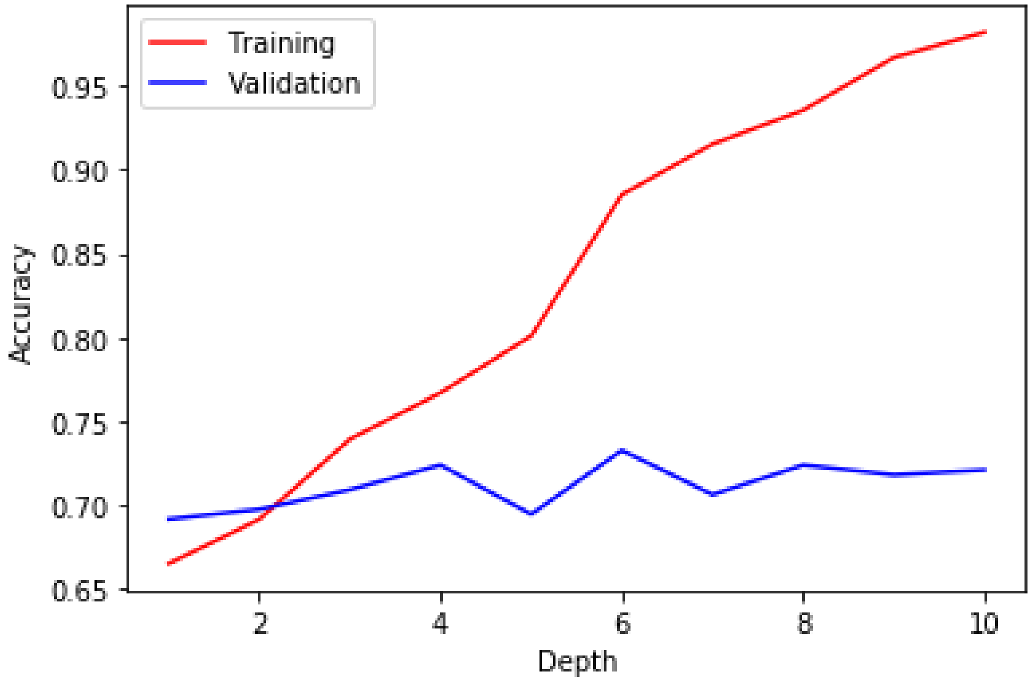
```
[0.6911764705882353, 0.6970588235294117, 0.7
088235294117647, 0.7235294117647059, 0.69411
76470588235, 0.7323529411764705, 0.705882352
9411765, 0.7235294117647059, 0.7176470588235
294, 0.7205882352941176]
```

In [24]:

```
x_range_decision_tree = list(range(1, 11))
plt.plot(x_range_decision_tree, training_depth_Accuracy,
color='r',label ='Training')
plt.plot(x_range_decision_tree, validation_depth_Accuracy
, color='b',label ='Validation')

plt.xlabel("Depth")
plt.ylabel("Accuracy")

plt.legend()
```

Out[24]:

```
<matplotlib.legend.Legend at 0xca4df0>
```



In [25]:

```
clf = DecisionTreeClassifier(criterion="gini", max_depth=
4,random_state=50)
clf.fit(X_train,y_train)

y_pred_train = clf.predict(X_test)
print(metrics.accuracy_score(y_test, y_pred_train))
```

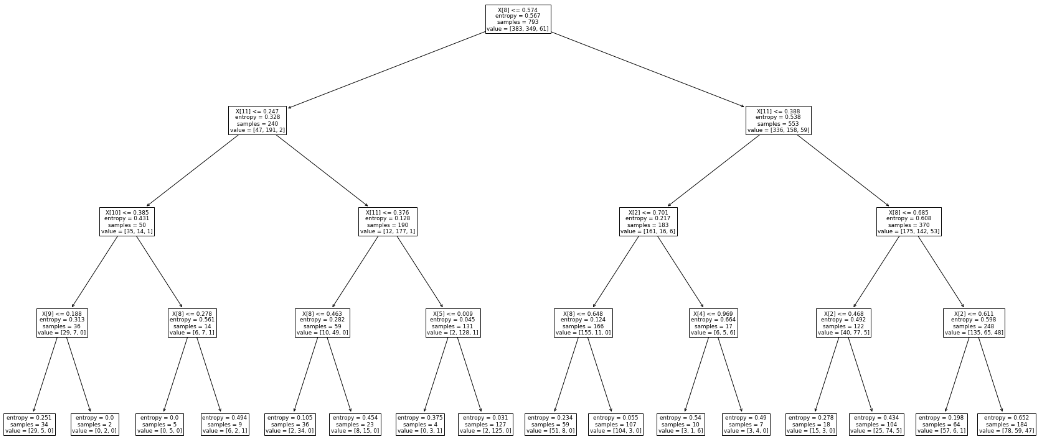
```
0.7235294117647059
```

In [26]:

```
plt.figure(figsize=(30,15))
from sklearn import tree
tree.plot_tree(clf)
```

Out[26]:

```
[Text(837.0, 733.86, 'X[8] <= 0.574\nentropy
= 0.567\nsamples = 793\nvalue = [383, 349, 6
1]'),
 Text(418.5, 570.78, 'X[11] <= 0.247\nentrop
y = 0.328\nsamples = 240\nvalue = [47, 191,
2]'),
 Text(209.25, 407.70000000000005, 'X[10] <=
0.385\nentropy = 0.431\nsamples = 50\nvalue
= [35, 14, 1]'),
 Text(104.625, 244.62, 'X[9] <= 0.188\nentro
py = 0.313\nsamples = 36\nvalue = [29, 7,
0]'),
 Text(52.3125, 81.54000000000008, 'entropy =
0.251\nsamples = 34\nvalue = [29, 5, 0]'),
 Text(156.9375, 81.54000000000008, 'entropy
= 0.0\nsamples = 2\nvalue = [0, 2, 0]'),
 Text(313.875, 244.62, 'X[8] <= 0.278\nentro
py = 0.561\nsamples = 14\nvalue = [6, 7,
1]'),
 Text(261.5625, 81.54000000000008, 'entropy
= 0.0\nsamples = 5\nvalue = [0, 5, 0]'),
 Text(366.1875, 81.54000000000008, 'entropy
= 0.494\nsamples = 9\nvalue = [6, 2, 1]'),
 Text(627.75, 407.70000000000005, 'X[11] <=
0.376\nentropy = 0.128\nsamples = 190\nvalue
= [12, 177, 1]'),
 Text(523.125, 244.62, 'X[8] <= 0.463\nentro
py = 0.282\nsamples = 59\nvalue = [10, 49,
0]'),
 Text(470.8125, 81.54000000000008, 'entropy
= 0.105\nsamples = 36\nvalue = [2, 34, 0]'),
 Text(575.4375, 81.54000000000008, 'entropy
= 0.454\nsamples = 23\nvalue = [8, 15, 0]'),
 Text(732.375, 244.62, 'X[5] <= 0.009\nentro
py = 0.045\nsamples = 131\nvalue = [2, 128,
1]'),
 Text(680.0625, 81.54000000000008, 'entropy
= 0.375\nsamples = 4\nvalue = [0, 3, 1]'),
 Text(784.6875, 81.54000000000008, 'entropy
= 0.031\nsamples = 127\nvalue = [2, 125,
0]'),
 Text(1255.5, 570.78, 'X[11] <= 0.388\nentro
py = 0.538\nsamples = 553\nvalue = [336, 15
8, 59]'),
 Text(1046.25, 407.70000000000005, 'X[2] <=
0.701\nentropy = 0.217\nsamples = 183\nvalue
= [161, 16, 6]'),
 Text(941.625, 244.62, 'X[8] <= 0.648\nentro
py = 0.124\nsamples = 166\nvalue = [155, 11,
0]'),
 Text(889.3125, 81.54000000000008, 'entropy
= 0.234\nsamples = 59\nvalue = [51, 8, 0]'),
 Text(993.9375, 81.54000000000008, 'entropy
= 0.055\nsamples = 107\nvalue = [104, 3,
0]'),
 Text(1150.875, 244.62, 'X[4] <= 0.969\nentr
opy = 0.664\nsamples = 17\nvalue = [6, 5,
6]'),
 Text(1098.5625, 81.54000000000008, 'entropy
= 0.54\nsamples = 10\nvalue = [3, 1, 6]'),
 Text(1203.1875, 81.54000000000008, 'entropy
= 0.49\nsamples = 7\nvalue = [3, 4, 0]'),
 Text(1464.75, 407.70000000000005, 'X[8] <=
0.685\nentropy = 0.608\nsamples = 370\nvalue
= [175, 142, 53]'),
 Text(1360.125, 244.62, 'X[2] <= 0.468\nentr
opy = 0.492\nsamples = 122\nvalue = [40, 77,
5]'),
 Text(1307.8125, 81.54000000000008, 'entropy
= 0.278\nsamples = 18\nvalue = [15, 3, 0]'),
 Text(1412.4375, 81.54000000000008, 'entropy
= 0.434\nsamples = 104\nvalue = [25, 74,
5]'),
 Text(1569.375, 244.62, 'X[2] <= 0.611\nentr
opy = 0.598\nsamples = 248\nvalue = [135, 6
5, 48]'),
 Text(1517.0625, 81.54000000000008, 'entropy
= 0.198\nsamples = 64\nvalue = [57, 6, 1]'),
 Text(1621.6875, 81.54000000000008, 'entropy
= 0.652\nsamples = 184\nvalue = [78, 59, 4
7]')]
```



In [27]:

```
conf_dec_tree = confusion_matrix(y_test, y_pred_train)
conf_dec_tree
```

Out[27]:

```
array([[150,  26,   1],
       [ 43,  94,   2],
       [ 20,   2,   2]], dtype=int64)
```

In [28]:

```
(conf_dec_tree[0][0] + conf_dec_tree[1][1] + conf_dec_tree[2][2]) /     np.sum(conf_dec_tree)
```

Out[28]:

```
0.7235294117647059
```

In [29]:

```
print("Accuracy", (conf_dec_tree[0][0] + conf_dec_tree[1][1]) / (conf_dec_tree[1][1] + conf_dec_tree[1][0] +

conf_dec_tree[0][1] + conf_dec_tree[0][0]))
print("Sensitivity", conf_dec_tree[1][1] / (conf_dec_tree[1][1] + conf_dec_tree[1][0] ))
print("Specificity", conf_dec_tree[0][0] / (conf_dec_tree[0][0] + conf_dec_tree[0][1] ))
print("Precision", conf_dec_tree[1][1] / (conf_dec_tree[1][1] + conf_dec_tree[0][1] ))
```

```
Accuracy 0.7795527156549521
Sensitivity 0.6861313868613139
Specificity 0.8522727272727273
Precision 0.7833333333333333
```

In [30]:

```
(conf_dec_tree[0][0] + conf_dec_tree[1][1]) / (conf_dec_tree[0][0] + conf_dec_tree[0][1] + conf_dec_tree[1][0] + conf_dec_tree[1][1])
```

Out[30]:

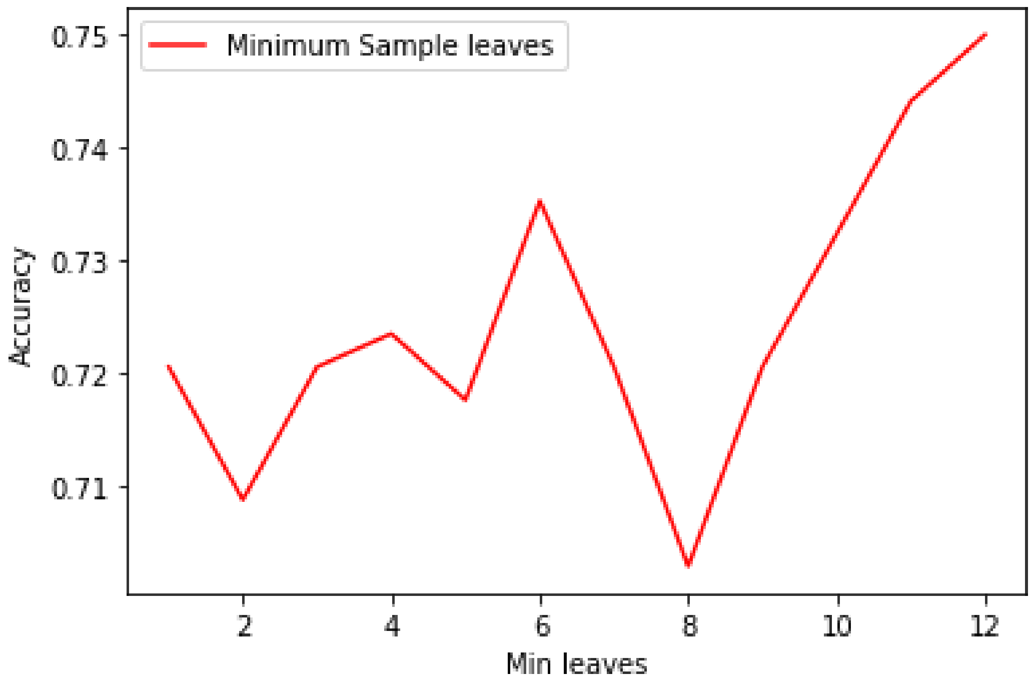
```
0.7795527156549521
```

In [31]:

```
listAccuracy = []
for i in range (0,12):
    clf = DecisionTreeClassifier(criterion="gini",random_state=50,max_depth=10, min_samples_leaf = i+1)
    clf.fit(X_train,y_train)
    y_pred_train = clf.predict(X_test)
    listAccuracy.append(metrics.accuracy_score(y_test, y_pred_train))
listAccuracy

x_range_decision_tree = list(range(1, 13))
plt.plot(x_range_decision_tree, listAccuracy, color='r',label ='Minimum Sample leaves')


plt.xlabel("Min leaves")
plt.ylabel("Accuracy")

plt.legend()
```

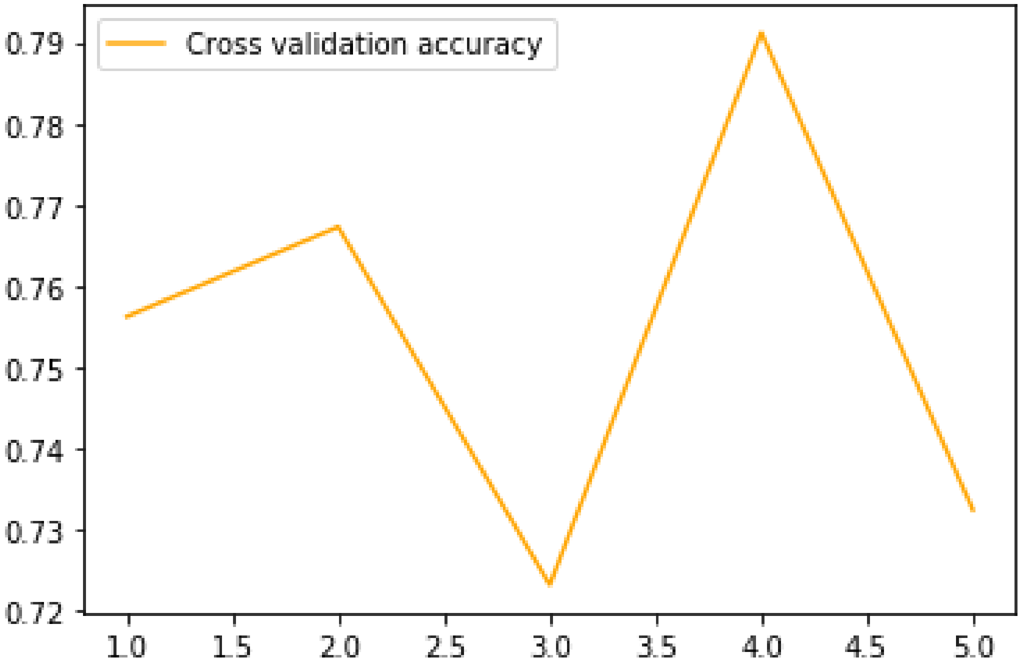
Out[31]:

```
<matplotlib.legend.Legend at 0xc61f70>
```


Minimum Sample leaves
0.75
0.74
0.73
0.72
0.71
Accuracy
2
4
6
8
10
12
Min leaves

In [32]:

```
clf = DecisionTreeClassifier(criterion="gini",random_stat
e=50,max_depth=4)
clf.fit(X_train,y_train)

scores = cross_val_score(clf, X_train, y_train, cv=5)


x_range = list(range(1, 6))
plt.plot(x_range, scores, color='orange',label ='Cross va
lidation accuracy')
plt.legend()
plt.show()
```



Boosting - Decision tree

In [33]:

```
# Create adaboost classifer object
no_of_esti = list(range(1, 11,1))
training_data_boosting_accuracy = []

for j in range(0, len(no_of_esti)):
    abc = GradientBoostingClassifier(n_estimators=30,rand
om_state=50,max_depth=no_of_esti[j])
    model = abc.fit(X_train, y_train)
    y_pred_train = model.predict(X_train)
    training_data_boosting_accuracy.append(metrics.accura
cy_score(y_train, y_pred_train))
```

In [34]:

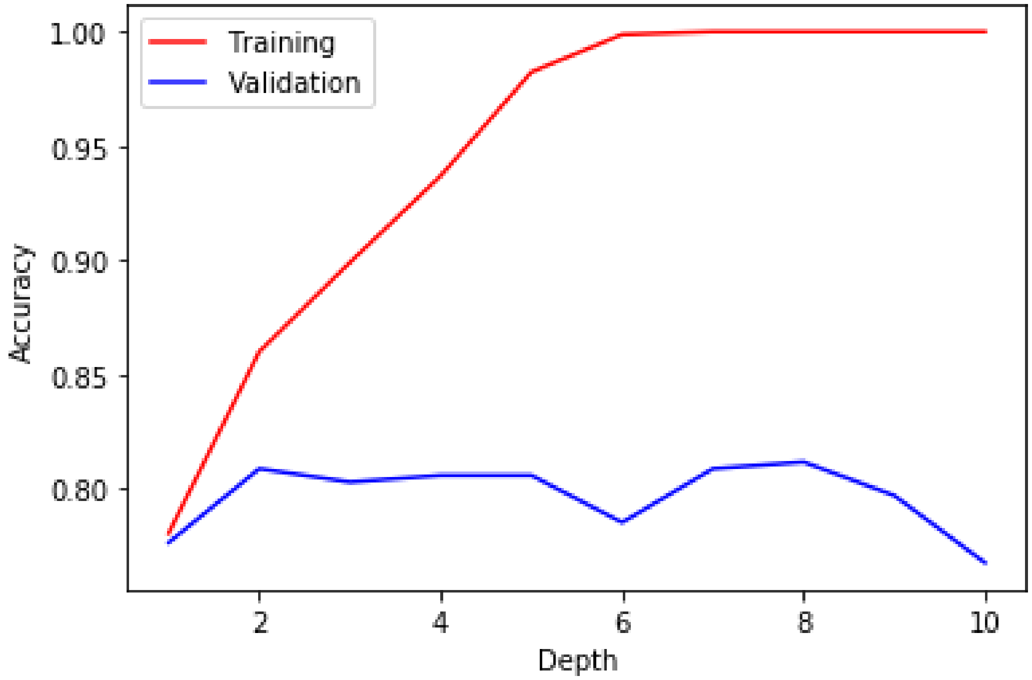
```
# Create adaboost classifer object
no_of_esti = list(range(1, 11,1))
validation_data_boosting_accuracy = []


for j in range(0, len(no_of_esti)):
    abc = GradientBoostingClassifier(n_estimators=50, ran
dom_state=50,max_depth=no_of_esti[j])
    model = abc.fit(X_train, y_train)
    y_pred_test = model.predict(X_test)
    validation_data_boosting_accuracy.append(metrics.accu
racy_score(y_test, y_pred_test))
```

In [35]:

```
x_range_decision_tree = list(range(1, 11))
plt.plot(x_range_decision_tree, training_data_boosting_accuracy, color='r',label ='Training')
plt.plot(x_range_decision_tree, validation_data_boosting_accuracy, color='b',label ='Validation')

plt.xlabel("Depth")
plt.ylabel("Accuracy")

plt.legend()
```

Out[35]:

```
<matplotlib.legend.Legend at 0xb3a0f0>
```



In [36]:

```
abc = GradientBoostingClassifier(max_depth=3, random_state=50)
model = abc.fit(X_train, y_train)
y_pred_test = model.predict(X_test)
cm = confusion_matrix(y_test, y_pred_test)
cm
```

Out[36]:

```
array([[141,  36,   0],
       [ 26, 113,   0],
       [  0,   0,  24]], dtype=int64)
```

In [37]:

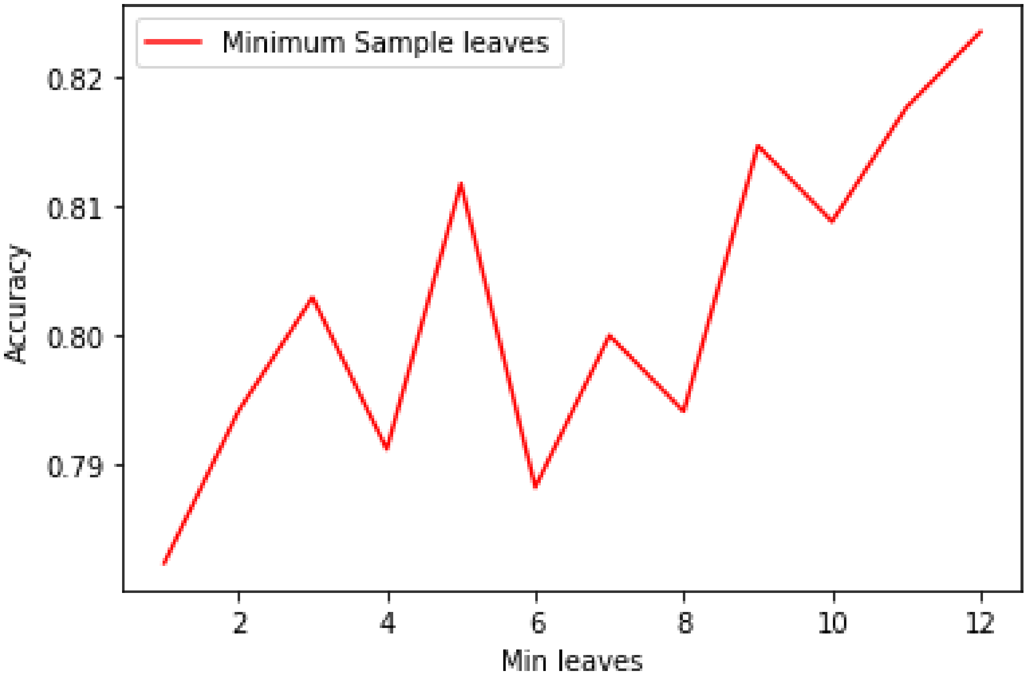
```
print("Accuracy", (cm[1][1] + cm[0][0]) / (cm[1][1] + cm[0][0] + cm[0][1] + cm[1][0] ) )
print("Sensitivity", cm[1][1] / (cm[1][1] + cm[1][0] ))
print("Specificity", cm[0][0] / (cm[0][0] + cm[0][1] ))
print("Precision", cm[1][1] / (cm[1][1] + cm[0][1] ))
```

```
Accuracy 0.8037974683544303
Sensitivity 0.8129496402877698
Specificity 0.7966101694915254
Precision 0.7583892617449665
```

In [38]:

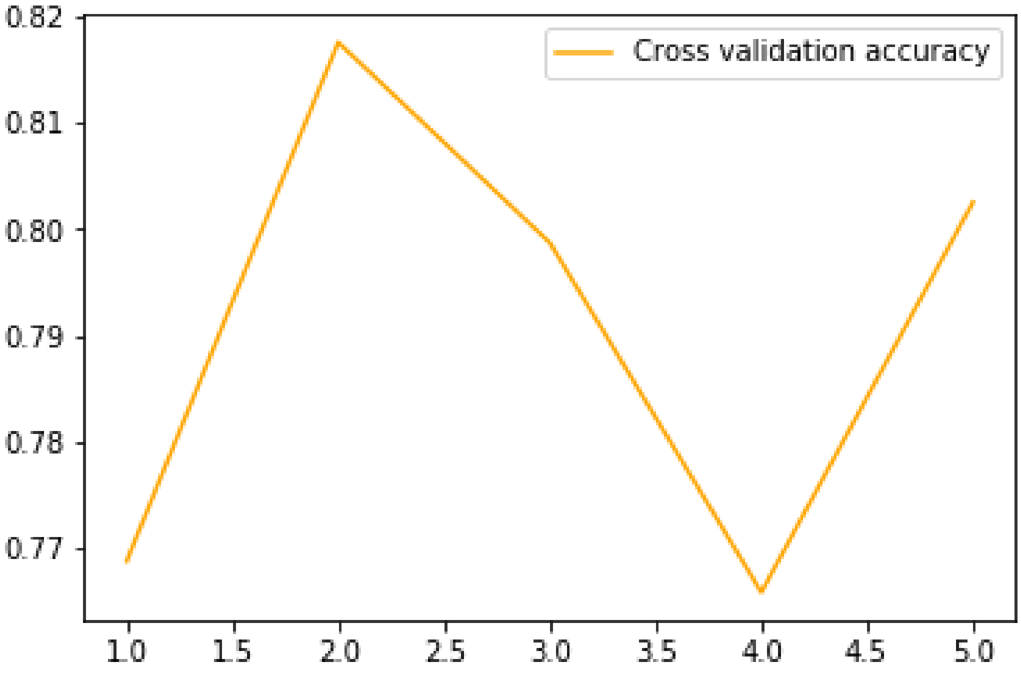
```
listAccuracy = []
for i in range (0,12):
    clf = GradientBoostingClassifier(random_state=50,max_
depth=10, min_samples_leaf = i+1)
    clf.fit(X_train,y_train)
    y_pred_train = clf.predict(X_test)
    listAccuracy.append(metrics.accuracy_score(y_test, y_
pred_train))
listAccuracy

x_range_decision_tree = list(range(1, 13))
plt.plot(x_range_decision_tree, listAccuracy, color='r',l
abel ='Minimum Sample leaves')



plt.xlabel("Min leaves")
plt.ylabel("Accuracy")

plt.legend()
```

```
Out[38]:

<matplotlib.legend.Legend at 0xa162b0>
```


Minimum Sample leaves
0.82
0.81
0.80
0.79
Accuracy
2
4
6
8
10
12
Min leaves

In [39]:

```
abc = GradientBoostingClassifier(max_depth=10, random_sta
te=50)
model = abc.fit(X_train, y_train)

scores = cross_val_score(model, X_train, y_train, cv=5)


x_range = list(range(1, 6))
plt.plot(x_range, scores, color='orange',label ='Cross va
lidation accuracy')
plt.legend()
plt.show()
```



In [ ]:

```
```