Support Vector Machine(SVM)

In [0]:
```
# importing important libraries
import numpy as np
import pandas as pd
import random
random.seed(3)
```

In [0]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/au
th?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.goog
leusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&r
esponse_type=code&scope=email%20https%3a%2f%2fwww.googleapis.com%2fa
uth%2fdocs.test%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20
https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20ht
tps%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly

Enter your authorization code:
··········
Mounted at /content/drive
```

In [0]:
```
df3 = pd.read_csv("/content/drive/My Drive/image classification projec
t/df3_final.csv",header=None) #importing data from directory
df3.head(5)
```

Out[0]:

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp	187	148	190	191	153	194
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp	25	14	23	66	40	56
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp	146	133	186	157	145	198
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp	27	16	31	70	55	86
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear	134	110	153	171	142	188

5 rows × 3080 columns

In [0]:
```
X = df3.iloc[:,7:3079].values
y = df3.iloc[:, 3079].values
```

In [0]:
```
from sklearn.model_selection import train_test_split
```

In [0]:
```
Xtr, Xte, ytr, yte = train_test_split(X, y, test_size=0.30)
#split data into training and testing group
```

In [0]:
```
from sklearn import svm
```

Linear kernel

In [0]:
```
classifier_linear = svm.SVC(kernel='linear')
#fit to the trainin data
classifier_linear.fit(Xtr,ytr)
```

Out[0]:
```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef
0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kerne
l='linear',
    max_iter=-1, probability=False, random_state=None, shrinking=Tru
e,
    tol=0.001, verbose=False)
```

In [0]:
```
y_pred = classifier_linear.predict(Xte)
```

In [0]:
```
print('accuracy: %f' % ( np.mean(y_pred == yte) ))
# Accuracy of the model
```

```
accuracy: 0.613311
```

In [0]:
```
from sklearn.metrics import classification_report, confusion_matrix
```

In [0]:
```
print(confusion_matrix(yte,y_pred))
```

```
[[  19   26   14    6    5   23    1]
 [  16   72   18    3   14   28    3]
 [   8   18  105    5   36  139    0]
 [   1    9    3    5    2   11    1]
 [  10   15   46    7   96  140    1]
 [  37   43  207    5  148 1610    8]
 [   0    6    2    1    7    6   19]]
```

In [0]:
```
print(classification_report(yte,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.21      0.20      0.21        94
           1       0.38      0.47      0.42       154
           2       0.27      0.34      0.30       311
           3       0.16      0.16      0.16        32
           4       0.31      0.30      0.31       315
           5       0.82      0.78      0.80      2058
           6       0.58      0.46      0.51        41

    accuracy                           0.64      3005
   macro avg       0.39      0.39      0.39      3005
weighted avg       0.66      0.64      0.65      3005
```

Polynomial Kernel

In [0]:
```
classifier_poly1 = svm.SVC(kernel='poly',degree=2)
#fit to the trainin data
classifier_poly1.fit(Xtr,ytr)
```

Out[0]:
```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef
0=0.0,
    decision_function_shape='ovr', degree=2, gamma='scale', kerne
l='poly',
    max_iter=-1, probability=False, random_state=None, shrinking=Tru
e,
    tol=0.001, verbose=False)
```

In [0]:
```
y_pred_poly1 = classifier_poly1.predict(Xte)
```

In [0]:
```
print('accuracy: %f' % ( np.mean(y_pred_poly1 == yte) ))
print(confusion_matrix(yte,y_pred_poly1))
print(classification_report(yte,y_pred_poly1))
```

```
accuracy: 0.731780
[[  15   25   12    2    8   32    0]
 [  11   72   14    1    5   48    3]
 [   4   12  100    1   22  172    0]
 [   2    9    1    3    3   14    0]
 [   4    9   50    2   71  178    1]
 [   8   18   57    2   48 1920    5]
 [   1    6    0    0    5   11   18]]
              precision    recall  f1-score   support

           0       0.33      0.16      0.22        94
           1       0.48      0.47      0.47       154
           2       0.43      0.32      0.37       311
           3       0.27      0.09      0.14        32
           4       0.44      0.23      0.30       315
           5       0.81      0.93      0.87      2058
           6       0.67      0.44      0.53        41

    accuracy                           0.73      3005
   macro avg       0.49      0.38      0.41      3005
weighted avg       0.69      0.73      0.70      3005
```

Radial basis function kernel

In [0]:
```
classifier_rbf = svm.SVC(kernel='rbf')
#fit to the trainin data
classifier_rbf.fit(Xtr,ytr)
```

Out[0]:
```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef
0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kerne
l='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=Tru
e,
    tol=0.001, verbose=False)
```

In [0]:
```
y_pred_rbf = classifier_rbf.predict(Xte)
```

In [0]:
```
print('accuracy: %f' % ( np.mean(y_pred_rbf== yte) ))
print(confusion_matrix(yte,y_pred_rbf))
print(classification_report(yte,y_pred_rbf))
```

```
accuracy: 0.725125
[[   4   21   16    0    2   51    0]
 [   3   48   13    0    2   88    0]
 [   0    7  102    0   11  191    0]
 [   2    7    3    0    1   19    0]
 [   1    2   34    0   23  255    0]
 [   2    8   32    0   14 2002    0]
 [   0    5    0    0    0   36    0]]
              precision    recall  f1-score   support

           0       0.33      0.04      0.08        94
           1       0.49      0.31      0.38       154
           2       0.51      0.33      0.40       311
           3       0.00      0.00      0.00        32
           4       0.43      0.07      0.13       315
           5       0.76      0.97      0.85      2058
           6       0.00      0.00      0.00        41

    accuracy                           0.73      3005
   macro avg       0.36      0.25      0.26      3005
weighted avg       0.65      0.73      0.66      3005


/usr/local/lib/python3.6/dist-packages/sklearn/metrics/_classificati
on.py:1272: UndefinedMetricWarning: Precision and F-score are ill-de
fined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

Sigmoid Kernel

In [0]:
```
classifier_sigmoid = svm.SVC(kernel='sigmoid')
#fit to the trainin data
classifier_sigmoid.fit(Xtr,ytr)
```

Out[0]:
```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef
0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kerne
l='sigmoid',
    max_iter=-1, probability=False, random_state=None, shrinking=Tru
e,
    tol=0.001, verbose=False)
```

In [0]:
```
y_pred_sigmoid = classifier_sigmoid.predict(Xte)
```

In [0]:
```
print('accuracy: %f' % ( np.mean(y_pred_sigmoid== yte) ))
print(confusion_matrix(yte,y_pred_sigmoid))
print(classification_report(yte,y_pred_sigmoid))
```

```
accuracy: 0.684859
[[   0    0    0    0    0   94    0]
 [   0    0    0    0    0  154    0]
 [   0    0    0    0    0  311    0]
 [   0    0    0    0    0   32    0]
 [   0    0    0    0    0  315    0]
 [   0    0    0    0    0 2058    0]
 [   0    0    0    0    0   41    0]]
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        94
           1       0.00      0.00      0.00       154
           2       0.00      0.00      0.00       311
           3       0.00      0.00      0.00        32
           4       0.00      0.00      0.00       315
           5       0.68      1.00      0.81      2058
           6       0.00      0.00      0.00        41

    accuracy                           0.68      3005
   macro avg       0.10      0.14      0.12      3005
weighted avg       0.47      0.68      0.56      3005


/usr/local/lib/python3.6/dist-packages/sklearn/metrics/_classificati
on.py:1272: UndefinedMetricWarning: Precision and F-score are ill-de
fined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```