Support Vector Machine (SVM)

Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Importing the dataset

```
dataset = pd.read_csv('Final1.csv')
X = dataset.iloc[:, 1:-1].values
y = dataset.iloc[:, -1].values
```

Splitting the dataset into the Training set and Test set

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state =
```

```
print(X_train)
```

```
[[7.8000e+02 1.0000e+00 2.0000e+00 ... 0.0000e+00 4.9230e+00 3.0332e+01]
 [8.9390e+03 4.0000e+00 2.7000e+01 ... 0.0000e+00 5.6781e+01 2.3801e+01]
 [1.8000e+02 1.0000e+00 9.0000e+00 ... 4.5240e+00 2.0475e+01 2.3095e+01]
 ...
 [       nan 1.0000e+00 1.2000e+01 ... 4.7370e+00 5.9690e+00 7.8863e+01]
 [4.7000e+03 0.0000e+00 3.0000e+00 ... 0.0000e+00 2.3427e+01 0.0000e+00]
 [5.0000e+03 4.0000e+00 1.8000e+01 ... 4.3050e+00 6.5186e+01 3.0341e+01]]
```

```
print(y_train)
```

```
[1 0 1 0 1 1 1 1 0 1 0 1 1 1 1 0 1 0 1 0 1 1 0 1 1 1 0 1 1 1 0 0 1 1 0]
```

```
print(X_test)
```

```
[[7.5000e+02 1.0000e+00 3.0000e+00 ... 5.7340e+00 4.9230e+00 2.4265e+01]
 [3.5000e+02 0.0000e+00 1.7000e+01 ... 4.7370e+00 3.1644e+01 1.8199e+01]
 [4.7700e+02 1.0000e+00 1.0000e+01 ... 5.3170e+00 1.2831e+01 2.4265e+01]
 ...
 [6.0000e+01 0.0000e+00 1.0000e+00 ... 1.1685e+01 0.0000e+00 0.0000e+00]
 [5.0000e+01 0.0000e+00 7.0000e+00 ... 0.0000e+00 2.3476e+01 2.4526e+01]
 [1.2500e+03 1.0000e+00 1.0000e+01 ... 1.0217e+01 3.0576e+01 5.3523e+01]]
```

```
print(y_test)
```

```
[1 0 0 1 1 1 0 0 1 1 1 0]
```

Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
print(X_train)
```

```
[[-0.80341904 -0.33859959 -1.23815074 ... -0.69816469 -1.13160888
   0.26643125]
 [ 1.55633291  1.63656468  2.22313327 ... -0.69816469  1.529547
   0.0048902 ]
 [-0.97695148 -0.33859959 -0.26899121 ...  0.30897241 -0.33353922
  -0.02338234]
 ...
 [        nan -0.33859959  0.14636287 ...  0.35639067 -1.07793212
   2.2099083 ]
 [ 0.33032623 -0.99698768 -1.09969938 ... -0.69816469 -0.18205378
  -0.94824692]
 [ 0.41709245  1.63656468  0.97707103 ...  0.26021843  1.96085973
   0.26679166]]
```

```
print(X_test)
```

```
[[-0.81209566 -0.33859959 -1.09969938 ...  0.57834374 -1.13160888
   0.02347159]
 [-0.92778396 -0.99698768  0.83861967 ...  0.35639067  0.2396115
  -0.21944803]
 [-0.89105292 -0.33859959 -0.13053985 ...  0.48551081 -0.72580031
   0.02347159]
 ...
 [-1.01165797 -0.99698768 -1.3766021  ...  1.90316091 -1.38423856
  -0.94824692]
 [-1.01455017 -0.99698768 -0.54589393 ... -0.69816469 -0.17953928
   0.03392362]
 [-0.6674853  -0.33859959 -0.13053985 ...  1.57635338  0.18480579
   1.19514025]]
```

Training the SVM model on the Training set

```
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0)
classifier.fit(X_train, y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
```

```
            max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
            verbose=False)
```

```
X_train = np.nan_to_num(X_train)
X_test = np.nan_to_num(X_test)
y_train = np.nan_to_num(y_train)
y_test = np.nan_to_num(y_test)
```

Predicting a new result

Predicting the Test set results

```
y_pred = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

```
     [[1 1]
      [0 0]
      [0 0]
      [1 1]
      [1 1]
      [1 1]
      [0 0]
      [1 0]
      [1 1]
      [1 1]
      [0 1]
      [0 0]]
```

Making the Confusion Matrix

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

```
     [[4 1]
      [1 6]]
     0.8333333333333334
```

✓ 0s completed at 9:44 PM