Support Vector Machine algorithm

February 17, 2022

[1]:
```
import pandas as pd
import numpy as np
from sklearn import datasets
from sklearn import svm
from sklearn import metrics
```

[2]:
```
clf_linear = svm.SVC(kernel='linear')
```

[3]:
```
cancer = datasets.load_breast_cancer()
li_ftrs = list(cancer.feature_names)
```

[8]:
```
def svm_fu(x):
    #seed = 23345
    rng = np.random.default_rng(x)
    idx_feat = (np.floor(30*rng.uniform(size=4))).astype(int)
    X = cancer["data"][:,idx_feat]
    label = cancer["feature_names"][idx_feat]
    Y = cancer["target"]
    model = clf_linear.fit(X,Y)
    selected_features = [li_ftrs[index] for index in idx_feat]
    print(selected_features)
    y_pred = clf_linear.predict(X)
    print("Accuracy:",metrics.accuracy_score(Y, y_pred))
    # Model Precision: what percentage of positive tuples are labeled as such?
    print("Precision:",metrics.precision_score(Y, y_pred))

    # Model Recall: what percentage of positive tuples are labelled as such?
    print("Recall:",metrics.recall_score(Y, y_pred))
```

[11]:
```
lst = []
n = int(input("Enter number of elements : "))
for i in range(0, n):
    ele = int(input())

    lst.append(ele)

print(lst)
for i in range(len(lst)):
```

```
    svm_fu(i)
```

```
Enter number of elements : 4
2223
123124
123124
13345
[2223, 123124, 123124, 13345]
['fractal dimension error', 'mean symmetry', 'mean texture', 'mean radius']
Accuracy: 0.8927943760984183
Precision: 0.8915343915343915
Recall: 0.9439775910364145
['compactness error', 'worst symmetry', 'mean smoothness', 'worst symmetry']
Accuracy: 0.6889279437609842
Precision: 0.6698113207547169
Recall: 0.9943977591036415
['mean concave points', 'mean symmetry', 'worst smoothness', 'mean perimeter']
Accuracy: 0.8910369068541301
Precision: 0.8851174934725848
Recall: 0.9495798319327731
['mean perimeter', 'mean concave points', 'worst smoothness', 'concave points
error']
Accuracy: 0.8910369068541301
Precision: 0.8851174934725848
Recall: 0.9495798319327731
```

```
[ ]:
```