Smart Systems Project

Nureen Ehab Mahmoud Mohamed Barakat

20221465124

Intelligent Systems

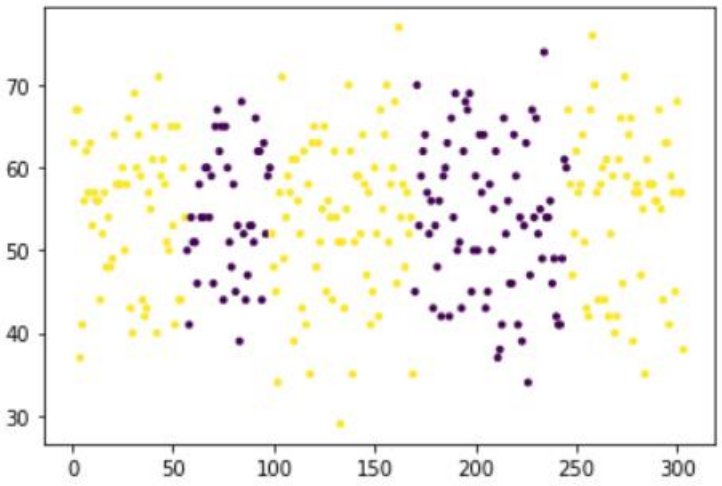
Report

SVMs are used in applications like classification
(support vector classification (SVC)).

The steps of SVC:

- **Import the dataset**
- **Explore the data to figure out what they look like**

  by plot as we show:



  we see that data can't be seperable by one line so we take that on count as while training SVM we must use RBF or polynomial kernel not linear

- **Pre-process the data**

  Scale the data by making features in range(0:1 or -1:1) to speed up training and for much accuracy

- **Split the data into attributes and labels**

  Attributes(training)=all coloumns except last one Labels(testing)=last column

  I use **train_test_split** function to split data with 0.25% testing , 0.75% training

- ## Train the SVM algorithm

  Define the model by **SVC** function and let **kernel** parameter be **rbf** (Gaussian Radial Basis Function) as data non linearly seperable

  Train the model by **fit** function and print train & test scores

- ## Make some predictions

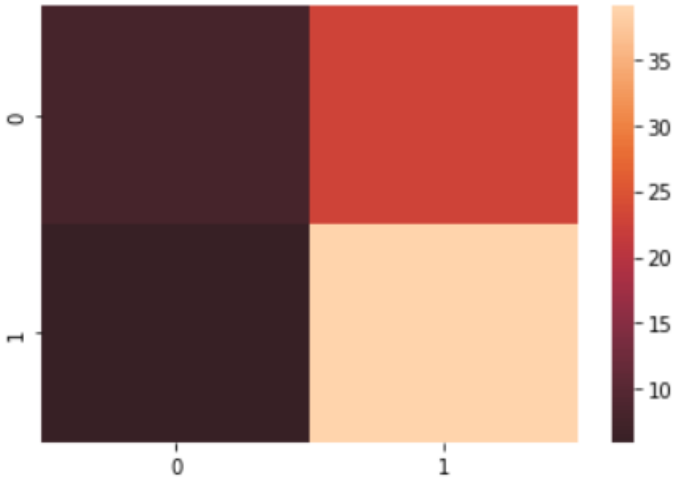
  predict values for SVC model by **predict** function and the predicted values of the provided test instances return in a form of an output of an array or sparse matrix.

```
predicted value for SVCModel is:  [1 1 0 0 1 1 1 0 1 0]
```

  Calculate the confusion matrix with **confusion_matrix** function to test the accuracy of prediction (comparing y actual with y predicted) as if main diagonal(TN,TP) has high values that means there is high accuracy

```
Confuison Matrix is:
 [[ 8 23]
 [ 6 39]]
```

  Plot the confusion matrix using **heatmap** to represent it in a 2-dim graph where the individual values that are contained in a matrix are represented as colours based on the intensity of the value



- **Evaluate the results of the algorithm**

  I will explain some of trails I made to evaluate the results:

  I change **rbf** kernel to **poly** kernel for trying achieve more accuracy but score of testing the model decrease so rbf kernel is better evaluation

  Result of **rbf** kernel:

```
SVCModel train score is:  0.8281938325991189
SVCModel test score is:  0.618421052631579
```

  Result of **poly** kernel:

```
SVCModel train score is:  0.8237885462555066
SVCModel test score is:  0.5526315789473685
```

  C and **gamma** are hypermeters which are set before the training model where "c" Smoothing coefficient used to control error and "Gamma" used to give curvature weight of the decision boundary.
  so we can change them to achieve more accuracy and evaluate the results:

  - Decreasing c leads to more accuracy and avoid overfitting.
  - When gamma is very small, the model is too constrained and cannot capture the complexity or "shape" of the data.

  But there is no thumb of rules that low C will work always or high C or medium C and the same for gamma , it totally depend upon data.

Note: Some of these steps can be combined depending on how we handle the data.

Reference of helper resources

https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/