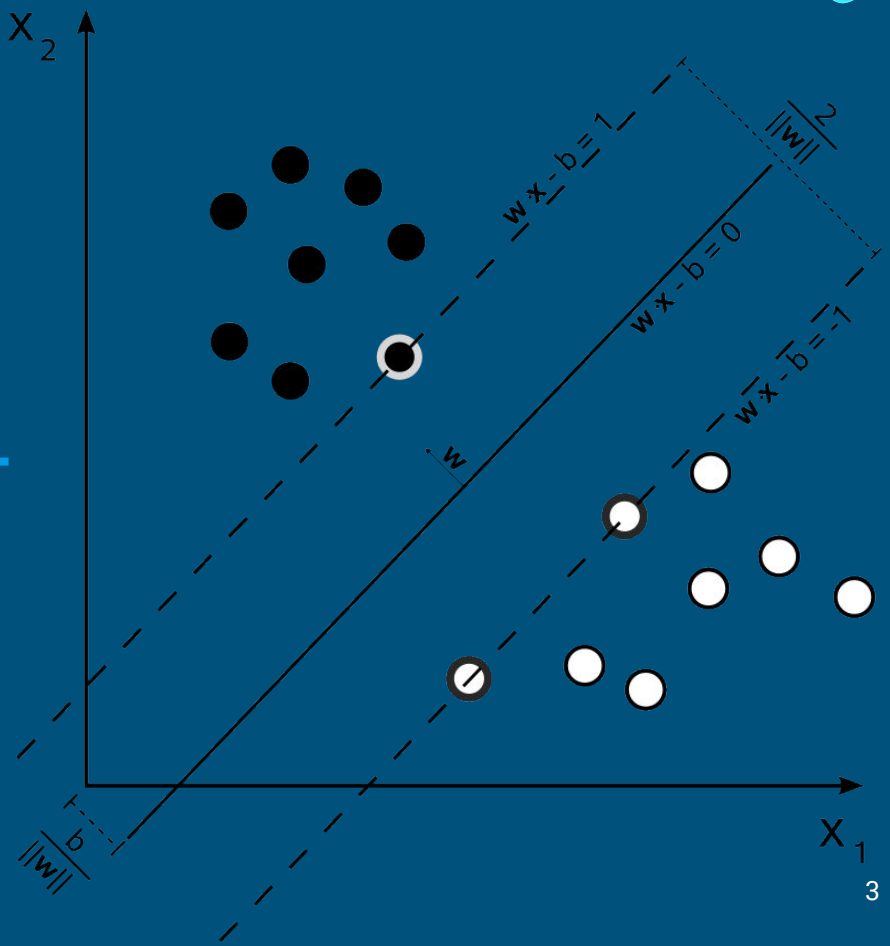
Machine Learning

By EvolkAI

Model Name:
Support Vector Classifier (SVC)

SVC aims to find the best possible line (or hyperplane) that can separate different classes of data. It does this by identifying a set of data points, called support vectors, that are closest to the boundary between the different classes.

SVC then uses these support vectors to create a decision boundary that maximizes the margin between the different classes of data, meaning the distance between the decision boundary and the closest data points. This maximization of the margin leads to a better generalization performance of the model.



```
import pandas as pd
data = pd.read_csv('iris_dataset.csv')
data.info()
```

In First line we Import pandas library as pd, then we read iris_dataset.csv file using the read_csv() function, and prints information about the data using the info() method.

```
feature = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']
predection_class = ['species']
X = data[feature].values
y = data[predection_class].values
```

This defines the ‘feature’ and ‘predection_class’ variables, which specify the columns of the data to use as ‘features’ and the column to use as the prediction target.
The code then creates ‘X’ and ‘y’ arrays containing the values of these columns from the data DataFrame.

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,y,test_size=0.30)
```

This imports the train_test_split function from the sklearn.model_selection module and uses it to split the data into training and testing sets. The test_size parameter specifies that 30% of the data should be used for testing.

```
print(f"Shape of X_test is {X_test.shape}")
print(f"Shape of X_train is {X_train.shape}")
print(f"Shape of Y_test is {Y_test.shape}")
print(f"Shape of Y_train is {Y_train.shape}")
```

These lines print the shapes of the training and testing data arrays.

OUTPUT

```
Shape of X_test is (45, 4)
Shape of X_train is (105, 4)
Shape of Y_test is (45, 1)
Shape of Y_train is (105, 1)
```

```
from sklearn import svm
clf = svm.SVC(kernel='linear')
clf.fit(X_train, Y_train)
Y_pred = clf.predict(X_test)
```

This imports the *svm* module from the sklearn library, creates an SVM classifier object with a linear kernel, fits the classifier to the training data using the *fit()* method, and uses the *predict()* method to generate predictions for the testing data.

```
from sklearn import metrics
print("Accuracy", metrics.accuracy_score(Y_test, Y_pred)*100)
```

This imports the metrics module from sklearn and uses the accuracy_score() function to calculate the accuracy of the model on the testing data. The result is printed to the console.

OUTPUT

```
Accuracy 97.77777777777777
```

Conclusion

The code trains a support vector machine (SVM) model to classify different species of iris flowers based on measurements of their sepal length, sepal width, petal length, and petal width. The accuracy of the model on the testing data is calculated to be printed to the console.

Therefore, the conclusion is that the SVM model with a linear kernel is able to accurately classify the species of iris flowers based on their measurements. The accuracy of the model can be used as a measure of its performance and can be compared to other models or approaches to evaluate their effectiveness in solving the same problem.

Thank You