sl-support-vector-mechanism-1

August 26, 2023

0.1 A.Vinay Kumar

0.2 21X05A6702

0.3 CSD(DATA SCIENCE)

0.4 NRCM

#Project Title: Using the support vector mechanism algorithm of supervised mechine learning,predict iris.csv dataset to findout the species will be same or different.

#Problem Statement: A American based Botanical garden grow iris flower in their labs but using bio technology in a single Tree different types of vareity flower is grow.Find out as a Data Science Enginner how much accuracy is there all category contain same species

#task 1: Preprocess the data in scikit.learn libery #Task 2: Load the data using sklearn model selection default argument #Task 3: On the basis of your dataset train test and split the SVM model #Task 4: Import support vector mechanism classifier.The SVM must be "Linear" #Task 5: Train the classifier on the training data #Task 6: Find out the prediction value on the test data #task 7: Test the model with the help of accuracy, should be lie in the range of 0-1

#Conclusion According to my support vector mechanism model the species are linear.With the accuracy of 1.00.

Hence proved model was successfully implement.

[1]:
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
```

[2]:
```
# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target
```

[3]:
```
# Consider only two classes for simplicity
X = X[y != 2]
y = y[y != 2]
```

[4]:
```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)
```

[5]:
```
# Create an SVM classifier
svm_classifier = SVC(kernel='linear')
```

[6]:
```
# Train the classifier on the training data
svm_classifier.fit(X_train, y_train)
```

[6]:
```
SVC(kernel='linear')
```

[7]:
```
# Make predictions on the test data
y_pred = svm_classifier.predict(X_test)
```

[8]:
```
# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

```
Accuracy: 1.00
```

[ ]:
```

```