sl-support-vector-mechanism-2

August 26, 2023

1 NAME DURGAPRASAD

2 21X05A6744

3 CSE(DATASCIENCE)

4 NARSIMHAREDDY ENGINEERING COLLEGE

5 Project title

USING THE SUPPORT VECTOR MECHANISM ALGORITHM OF SUPERVISED MACHINE 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

6 Conclusion

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

Hence proved model was successfully implement.

[3]:

```
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
```

[4]:

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

[5]:

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

[6]:

```
# 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)
```

[11]:

```
#Create an SVM classifier
svm_classifier = SVC(kernel='linear')
```

[8]:

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

[8]:

```
SVC(kernel='linear')
```

[9]:

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

[10]:

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

```
Accuracy: 1.00
```

[ ]:

```

```