

# sl-support-vector-mechanism-2

August 26, 2023

#Name : Algam Sai Krishna #Roll No: 21X05A6703 #Branch :Data Science #College: Narsimha Reddy Engineering College

#Project Title: Using the support vector mechanism algorithm of supervised machine learning, predict iris.csv dataset to find out 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 variety flower is grow. Find out as a Data Science Engineer how much accuracy is there all category contain same species

#Task 1: Preprocess the data in scikit.learn library #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')
```

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

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

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

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

Accuracy: 1.00

```
[ ]:
```