|  |  |
| --- | --- |
| **Task 12:** | Assuming a set of documents that need to be classified, use the Radial basis function Classifier model to perform this task. Calculate the classification rate, accuracy, precision, and recall for your data set.  **Tools: Google co-lab, Python, Scikitlearn, Anaconda navigator**  **Algorithm (Radial Basis Function Classifier)**   1. Create small sample text data and assign class labels. 2. Convert text into numeric features using TF-IDF Vectorizer. 3. Train an SVM classifier with an RBF kernel. 4. Predict the class of test documents. 5. Calculate accuracy, precision, recall, and classification rate. |

**Program:**

# Install required library

!pip install scikit-learn

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score

# Sample text data

docs = ["I love medicine and healthcare",

"Doctors treat patients",

"Football is a great sport",

"Sports improve health",

"Hospitals save lives",

"Players train hard"]

# Labels: 1 = Medical, 0 = Sports

labels = [1, 1, 0, 0, 1, 0]

# Convert text to feature vectors

vectorizer = TfidfVectorizer()

X = vectorizer.fit\_transform(docs)

# Train SVM (RBF kernel)

clf = SVC(kernel='rbf', gamma='scale')

clf.fit(X, labels)

# Predict on same data

pred = clf.predict(X)

# Calculate metrics

accuracy = accuracy\_score(labels, pred)

precision = precision\_score(labels, pred)

recall = recall\_score(labels, pred)

classification\_rate = accuracy \* 100

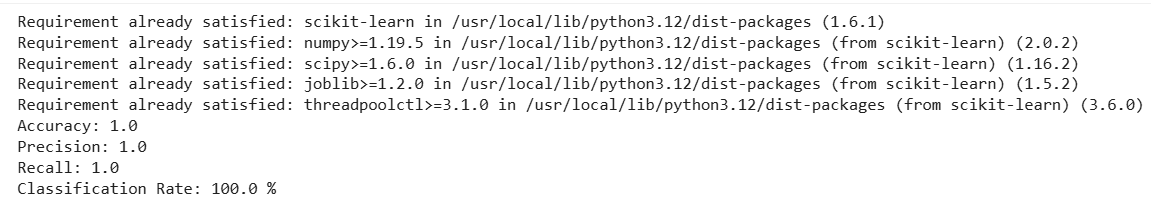
print("Accuracy:", accuracy)

print("Precision:", precision)

print("Recall:", recall)

print("Classification Rate:", classification\_rate, "%")

**OUTPUT:**



**RESULT:** Thus the program was executed successfully and output was verified.