**IMPLEMENTATION OF CLASSIFICATION ALGORITHMS**

Ex. No. 4

**Aim:**

To implement the Support Vector Machine (SVM) algorithm for classify the types of iris using pandas and numpy.

**Algorithm:**

Iris Species Classification using SVM

1. Import required libraries:

- Google Colab drive library for mounting Google Drive

- pandas for data manipulation

- numpy for numerical operations

- seaborn and matplotlib.pyplot for data visualization

- sklearn for datasets and model selection

2. Mount Google Drive to access the dataset file.

3. Load the Iris dataset from Google Drive into a pandas DataFrame.

4. Encode the categorical "Species" column to numerical values:

- Iris-setosa: 1

- Iris-versicolor: 2

- Iris-virginica: 3

5. Ignore warnings to ensure a clean output.

6. Check and print the column names of the dataset.

7. Replace the "Species" column with encoded numerical values.

8. Convert the "Species" column to numeric data type.

9. Select only the numeric columns from the dataset.

10. Compute the correlation matrix for the numeric columns.

11. Visualize the correlation matrix using a heatmap.

12. Split the dataset into features (X) and target (y):

- X contains all columns except the last one

- y contains the last column (Species)

13. Split the dataset into training and testing sets using train\_test\_split:

- Training set: 75% - Testing set: 25% - Random state: 0

14. Create an SVM classifier with a linear kernel and random state as 0.

15. Fit the SVM classifier to the training data.

16. Make predictions on the testing data.

17. Compute the confusion matrix to evaluate the predictions.

18. Perform 10-fold cross-validation to evaluate the model's accuracy:

- Compute mean accuracy

- Compute standard deviation of accuracy

19. Print the confusion matrix, mean accuracy, and standard deviation of accuracy.

20. End Algorithm.

**Code:**

from google.colab import drive

drive.mount('/content/gdrive',force\_remount=True)

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn import datasets, model\_selection

iris = pd.read\_csv("/content/gdrive/MyDrive/Colab Notebooks/Iris.csv")

#Encoding the categorical column

iris = iris.replace({"class": {"Iris-setosa":1,"Iris-versicolor":2, "Iris-virginica":3}})

#Visualize the new dataset

iris.head()

import warnings

warnings.filterwarnings('ignore')

# Check the column names

print(iris.columns)

# Encoding the categorical column

# Using the correct column name "Species"

iris = iris.replace({"Species": {"Iris-setosa": 1, "Iris-versicolor": 2, "Iris-virginica": 3}})

# Convert the "Species" column to numeric

iris["Species"] = pd.to\_numeric(iris["Species"])

# Select only numeric columns

numeric\_cols = iris.select\_dtypes(include=[np.number]).columns.tolist()

# Compute the correlation matrix

corr\_matrix = iris[numeric\_cols].corr()

# Plot the heatmap

plt.figure(figsize=(8, 6))

sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation on Iris Species')

plt.show()

X = iris.iloc[:,:-1]

y = iris.iloc[:, -1].values

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 0)

#Create the SVM model

from sklearn.svm import SVC

classifier = SVC(kernel = 'linear', random\_state = 0)

#Fit the model for the data

classifier.fit(X\_train, y\_train)

#Make the prediction

y\_pred = classifier.predict(X\_test)

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

print(cm)

from sklearn.model\_selection import cross\_val\_score

accuracies = cross\_val\_score(estimator = classifier, X = X\_train, y = y\_train, cv = 10)

print("Accuracy: {:.2f} %".format(accuracies.mean()\*100))

print("Standard Deviation: {:.2f} %".format(accuracies**.**std()\*100))

**Output:**

i)

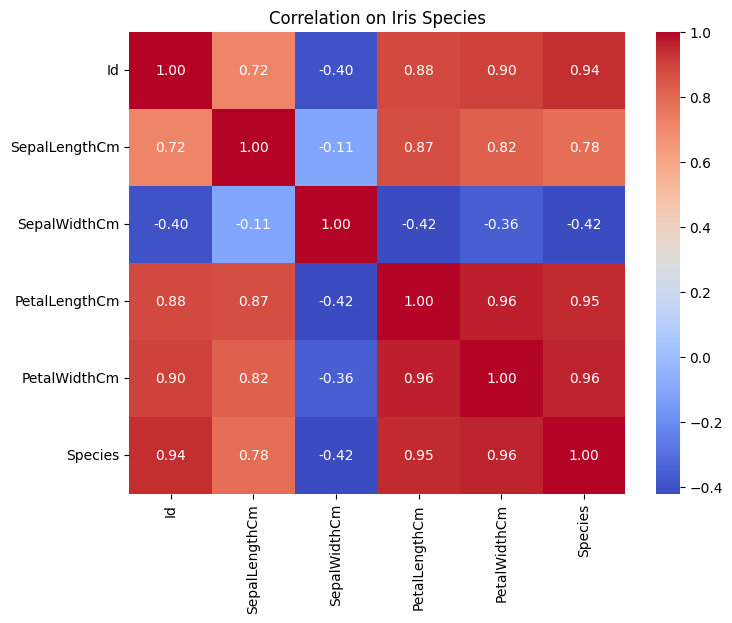
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **SepalLengthCm** | **SepalWidthCm** | **PetalLengthCm** | **PetalWidthCm** | **Species** |
| 0 | 1 | 5.1 | 1.4 | 0.2 | Iris-setosa |
| 1 | 2 | 4.9 | 1.4 | 0.2 | Iris-setosa |
| 2 | 3 | 4.7 | 1.3 | 0.2 | Iris-setosa |
| 3 | 4 | 4.6 | 1.5 | 0.2 | Iris-setosa |
| 4 | 5 | 5.0 | 1.4 | 0.2 | Iris-setosa |

ii)

Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',

'Species'],

dtype='object')



Confesion Matrix: [[13 0 0]

[ 0 16 0]

[ 0 0 9]]

Accuracy: 100.00 %

Standard Deviation: 0.00 %

**Rubrics:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Problem**  **Understanding**  **(10)** | **Implementation**  **(20)** | **Viva**  **(10)** | **Time Management**  **(10)** | **Total**  **(50)** |
|  |  |  |  |  |

**Result:**

Thus the implementation of SVM algorithm for classifying the type of iris using numpy and pandas was successfully executed and he output was verified.