**Practical: 7**

**Aim: Write a python program to classify various types of from iris dataset using support vector machines (SVM)**

**Code:**

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

from sklearn.metrics import confusion\_matrix

from sklearn.svm import SVC

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

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Define the col names

colnames = ["sepal\_length\_in\_cm", "sepal\_width\_in\_cm",

"petal\_length\_in\_cm", "petal\_width\_in\_cm", "class"]

# Read the dataset

dataset = pd.read\_csv(

"https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data", header=None, names=colnames)

# Data

print(dataset.head())

# Encoding the categorical column

dataset = dataset.replace(

{"class": {"Iris-setosa": 1, "Iris-versicolor": 2, "Iris-virginica": 3}})

# Visualize the new dataset

print(dataset.head())

# Plot the dataset

plt.figure(1)

sns.heatmap(dataset.corr())

plt.title('Correlation On iris Classes')

plt.show()

# Train and Test split

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

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, y, test\_size=0.25, random\_state=0)

# Create the SVM model

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)

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

print(cm)

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))

Data Head

Text

Description automatically generated with medium confidence

Data Head after replacing class values with 1,2,3

A picture containing diagram

Description automatically generated

A picture containing background pattern

Description automatically generated

Correlation Matrix

A picture containing icon

Description automatically generated

Accuracy and Standard deviation

