|  |  |
| --- | --- |
| **Ex. No: 9**  **Date: 21.03.24** | **SUPPORT VECTOR REGRESSION** |

**Name: S Sowmithaa Sri**

**Reg no: 22011101096**

**Aim:**

To choose a prediction dataset from challenge website. Apply SVR by using different kernels, fine tune hyperparameters. Compare best prediction result with that of linear regression and write your observations.

**Dataset description:**

The Wine Quality dataset contains chemical properties of wines along with their quality ratings.

**Features (input variables):**

Fixed acidity: The amount of fixed acids in the wine (g/dm³).

Volatile acidity: The amount of volatile acids in the wine (g/dm³).

Citric acid: The amount of citric acid in the wine (g/dm³).

Residual sugar: The amount of residual sugar in the wine (g/dm³).

Chlorides: The amount of chlorides in the wine (g/dm³).

Free sulfur dioxide: The amount of free sulfur dioxide in the wine (mg/dm³). Total sulfur dioxide: The amount of total sulfur dioxide in the wine (mg/dm³).

Density: The density of the wine (g/cm³).

pH: The pH level of the wine.

Sulphates: The amount of sulphates in the wine (g/dm³).

Alcohol: The alcohol content of the wine (% by volume).

**Target (output variable):**

Quality: The quality rating of the wine, ranging from 0 to 10.

The dataset contains 1599 samples.

**Code:**

import pandas as pd

import numpy as np

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

from sklearn.preprocessing import StandardScaler

from sklearn.svm import SVR

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

from sklearn.datasets import load\_wine

# Load the Wine Quality dataset

wine = load\_wine(as\_frame=True)

X = wine.data

y = wine.target

# Splitting the dataset into training and testing sets and feature scaling

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Support Vector Regression with different kernels

kernels = ['linear', 'poly', 'rbf', 'sigmoid']

best\_svr = None

best\_svr\_rmse = float('inf')

best\_svr\_mse = float('inf')

best\_kernel = None

for kernel in kernels:

svr = SVR(kernel=kernel)

parameters = {'C': [0.1, 1, 10, 100], 'gamma': [0.1, 0.01, 0.001, 0.0001]}

grid\_search = GridSearchCV(estimator=svr, param\_grid=parameters, scoring='neg\_mean\_squared\_error', cv=5)

grid\_search.fit(X\_train\_scaled, y\_train)

svr\_mse = -grid\_search.best\_score\_

svr\_rmse = np.sqrt(svr\_mse)

print(f"MSE for {kernel} kernel:", svr\_mse)

if svr\_mse < best\_svr\_mse:

best\_svr\_rmse = svr\_rmse

best\_svr = grid\_search.best\_estimator\_

best\_svr\_mse = svr\_mse

best\_kernel = kernel

# Linear Regression

lr = LinearRegression()

lr.fit(X\_train\_scaled, y\_train)

lr\_pred = lr.predict(X\_test\_scaled)

lr\_mse = mean\_squared\_error(y\_test, lr\_pred)

lr\_rmse = np.sqrt(lr\_mse)

# Output results

print("\nBest SVR Kernel:", best\_kernel)

print("Best SVR MSE:", best\_svr\_mse)

print("Linear Regression RMSE:", lr\_rmse)

print("Linear Regression MSE:", lr\_mse)

**Result:**

Best Parameters: {'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}

precision recall f1-score support

0 1.00 1.00 1.00 33

1 1.00 1.00 1.00 28

2 1.00 1.00 1.00 33

3 1.00 0.97 0.99 34

4 1.00 1.00 1.00 46

5 0.98 0.98 0.98 47

6 0.97 1.00 0.99 35

7 0.97 0.97 0.97 34

8 1.00 1.00 1.00 30

9 0.97 0.97 0.97 40

accuracy 0.99 360

macro avg 0.99 0.99 0.99 360

weighted avg 0.99 0.99 0.99 360

**Result Description:**

1.Best SVR Kernel and MSE: - The analysis was performed using Support Vector Regression (SVR) with different kernels: linear, polynomial, radial basis function (rbf), and sigmoid.

- For each kernel, a grid search with cross-validation was conducted to fine-tune the hyperparameters (C and gamma).

- The kernel with the lowest mean squared error (MSE) on the training set was selected as the best performing kernel.

- The best SVR kernel chosen along with its corresponding MSE is provided in the output.

2. Linear Regression RMSE and MSE:

- Linear Regression was applied as a baseline model for comparison.

- The mean squared error (MSE) and root mean squared error (RMSE) were calculated for Linear Regression on the test set.

3. Observations:

- By comparing the MSE of the best SVR model with that of Linear Regression, we can observe which model performs better in terms of prediction accuracy.

- Additionally, comparing the RMSE of Linear Regression with that of the best SVR model gives us an insight into the effectiveness of SVR in capturing the underlying patterns in the data.

**Conclusion:**

Thus, the implementation of SVR is executed successfully and the output is verified.