

SVM

Name:Nagaveni L G

SRN : PES2UG21CS315

SEC:5F

```
from sklearn.svm import SVC, SVR
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
import pandas as pd
import numpy as np
from collections import namedtuple

Dataset = namedtuple("Dataset", ["X", "y"])

class DatasetHelper:
    def read(self, dataset_path):
        data = pd.read_csv(dataset_path)

        # X -> Contains the features
        X = data.iloc[:, 0:-1]
        # y -> Contains all the targets
        y = data.iloc[:, -1]

        dataset = Dataset(X, y)
        return dataset

class SVM:

    def __init__(self, datasets):
        self.datasets = datasets
        self.models = []

    def train_model(self, model, dataset):
        if model:
            X = dataset.X
            y = dataset.y
            model.fit(X, y)

    def build_models(self):
        """
        You are required to define 3 SVM models in this function. Only define
        them, code for loading the corresponding
        datasets and training the models is pre-written.
```

Model 1: Regression model trained on dataset 1 (train1.csv). This model will be tested on hidden test datasets based on which marks will be awarded.

Model 2: Classification model trained on dataset 2 (train2.csv). This model will be tested on hidden test datasets based on which marks will be awarded.

Model 3: Classification model trained on dataset 3 (train3.csv) with spiral data distribution. This model will be tested on a visible test dataset (test3_visible.csv), based on which marks will be awarded.

HINT: Try experimenting with various hyperparameters and keep kernel trick in mind. This is a difficult dataset and high accuracies are not expected.

General Instructions:

Stick to using sklearn's SVM module only to define the models.

You are free to use any pre-processing you wish to use

Note: Use the sklearn Pipeline to add the pre-processing as a step in the model pipeline

Stick to using sklearn Pipeline only and not any other custom Pipeline to add preprocessing

```
"""
model1 = Pipeline([
    ('scaler', StandardScaler()),
    ('svm_regressor', SVR(kernel='linear', C=0.5)) # You can change
the kernel as needed
])

model2 = Pipeline([
    ('scaler', StandardScaler()),
    ('svm_classifier', SVC(kernel='rbf', C=1.0, gamma=0.1)) # Adjust
the kernel and C value
])

model3 = Pipeline([
    ('scaler', StandardScaler()),
    ('svc', SVC(kernel='rbf', C=10.5, gamma=10.5)) # Example: Trying a
Polynomial kernel
])

self.models.extend([model1, model2, model3])
assert len(self.models) == len(self.datasets), \
    f"Number of models {len(self.models)} is not the same as the
number of datasets {len(self.datasets)}"
```

```
for i in range(len(self.models)):
    self.train_model(self.models[i], self.datasets[i])
```

Output:

```
● PS C:\Users\Praka\OneDrive\Documents\5thSem\MI\SVM (students)> python test.py --ID EC_F_PES2UG21CS315
MSE: 0.002
Accuracy: 95.56%
Accuracy: 74.36%
○ PS C:\Users\Praka\OneDrive\Documents\5thSem\MI\SVM (students)> █
```