

Machine Learning Lab 2

Akhilendra Pratap 211112438

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
import os

train_dirs=[]
test_dirs=[]
for dir in os.listdir("./"):
    if(dir.find("5-fold")!=-1):
        train_dirs.append("./"+dir+"/train/")
        test_dirs.append("./"+dir+"/test/")

def cal_header_val(file_path):
    with open(file_path, "r") as file:
        lines=file.readlines()
    return lines.index('@data\n')+1

headers=[]
for dir in train_dirs:
    file_path=dir+os.listdir (dir)[0]
    headers.append(cal_header_val(file_path))

headers=np.array(headers)
```

SVM

```
def Support_Vector_Regression(train_file, test_file, header):
    train_df = pd.read_csv(train_file, header=header, delimiter=",")
    test_df = pd.read_csv(test_file, header=header, delimiter=",")

    X_train = train_df.iloc[:, :-1].values
    y_train = train_df.iloc[:, -1].values

    X_test = test_df.iloc[:, :-1].values
    y_test = test_df.iloc[:, -1].values

    from sklearn.svm import SVR
    width=np.array([2**i for i in range(-18, 21, 2)])

    best_mse, best_w=float('inf'), None

    for w in width:
        regressor=SVR(kernel="rbf", gamma=w)
        regressor.fit(X_train, y_train)
        y_pred=regressor.predict(X_test)
```

```

        from sklearn.metrics import mean_squared_error
        mse=mean_squared_error(y_pred, y_test)
        if(mse<best_mse):
            best_mse, best_w=mse, w

    return np.array([best_mse, best_w])

for train_dir, test_dir, header in zip(train_dirs, test_dirs,
headers):
    train_files=os.listdir(train_dir)
    test_files=os.listdir(test_dir)
    val=np.zeros(2)
    for train, test in zip(train_files, test_files):
        val+=Support_Vector_Regression(train_dir+train, test_dir+test,
header)
    print(train_dir)
    val/=len(train_files)
    val[0]=math.sqrt(val[0])
    val=pd.DataFrame(val, index=["RMSE", "Alpha"], columns=["Values"])
    print(val)
    print("-----\n")

```

/usr/lib/python3/dist-packages/scipy/__init__.py:146: UserWarning: A NumPy version >=1.17.3 and <1.25.0 is required for this version of SciPy (detected version 1.26.3

warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}")

./ele-1-5-fold/train/

	Values
RMSE	1139.563045
Alpha	0.000015

./diabetes-5-fold/train/

	Values
RMSE	0.295963
Alpha	4.450000

./quake-5-fold/train/

	Values
RMSE	0.136809
Alpha	4313.800000

./laser-5-fold/train/

	Values
RMSE	23.374021
Alpha	0.000098

```

-----
./plastic-5-fold/train/
      Values
RMSE    1.490691
Alpha   0.010937
-----

```

SVM / MAE

```

def Support_Vector_Regression(train_file, test_file, header):
    train_df = pd.read_csv(train_file, header=header, delimiter=",")
    test_df = pd.read_csv(test_file, header=header, delimiter=",")

    X_train = train_df.iloc[:, :-1].values
    y_train = train_df.iloc[:, -1].values

    X_test = test_df.iloc[:, :-1].values
    y_test = test_df.iloc[:, -1].values

    from sklearn.svm import SVR
    width=np.array([2**i for i in range(-18, 21, 2)])

    best_mse, best_w=float('inf'), None

    for w in width:
        regressor=SVR(kernel="rbf", gamma=w)
        regressor.fit(X_train, y_train)
        y_pred=regressor.predict(X_test)
        from sklearn.metrics import mean_absolute_error
        mse=mean_absolute_error(y_pred, y_test)
        if(mse<best_mse):
            best_mse, best_w=mse, w

    return np.array([best_mse, best_w])

for train_dir, test_dir, header in zip(train_dirs, test_dirs,
headers):
    train_files=os.listdir(train_dir)
    test_files=os.listdir(test_dir)
    val=np.zeros(2)
    for train, test in zip(train_files, test_files):
        val+=Support_Vector_Regression(train_dir+train, test_dir+test,
header)
    print(train_dir)
    val/=len(train_files)
    val=pd.DataFrame(val, index=["MAE", "Alpha"], columns=["Values"])
    print(val)
    print("-----\n")

```

```

./ele-1-5-fold/train/
      Values
MAE    792.005959
Alpha   0.000015
-----

./diabetes-5-fold/train/
      Values
MAE     0.183959
Alpha   4.450000
-----

./quake-5-fold/train/
      Values
MAE      0.103932
Alpha  1228.800002
-----

./laser-5-fold/train/
      Values
MAE    12.226322
Alpha   0.000244
-----

./plastic-5-fold/train/
      Values
MAE     1.117342
Alpha   0.071875
-----

```

SVM / MSE

```

def Support_Vector_Regression(train_file, test_file, header):
    train_df = pd.read_csv(train_file, header=header, delimiter=",")
    test_df = pd.read_csv(test_file, header=header, delimiter=",")

    X_train = train_df.iloc[:, :-1].values
    y_train = train_df.iloc[:, -1].values

    X_test = test_df.iloc[:, :-1].values
    y_test = test_df.iloc[:, -1].values

    from sklearn.svm import SVR
    width=np.array([2**i for i in range(-18, 21, 2)])

    best_mse, best_w=-float('inf'), None

    for w in width:
        regressor=SVR(kernel="rbf", gamma=w)

```

```

        regressor.fit(X_train, y_train)
        y_pred=regressor.predict(X_test)
        from sklearn.metrics import r2_score
        mse=r2_score(y_pred, y_test)
        if(mse>best_mse):
            best_mse, best_w=mse, w

    return np.array([best_mse, best_w])

for train_dir, test_dir, header in zip(train_dirs, test_dirs,
headers):
    train_files=os.listdir(train_dir)
    test_files=os.listdir(test_dir)
    val=np.zeros(2)
    for train, test in zip(train_files, test_files):
        val+=Support_Vector_Regression(train_dir+train, test_dir+test,
header)
    print(train_dir)
    val/=len(train_files)
    val=pd.DataFrame(val, index=["R2 score", "Alpha"],
columns=["Values"])
    print(val)
    print("-----\n")

./ele-1-5-fold/train/
      Values
R2 score -312.579759
Alpha      0.000015
-----

./diabetes-5-fold/train/
      Values
R2 score  0.755725
Alpha     55.450000
-----

./quake-5-fold/train/
      Values
R2 score  -1.180579
Alpha     460.806250
-----

./laser-5-fold/train/
      Values
R2 score  0.417075
Alpha     0.000098
-----

./plastic-5-fold/train/
      Values

```

R2 score 0.782111

Alpha 0.015625
