[**Total Points: 25**]

Support Vector Classification

Objective:

The objective of this assignment is to implement an image classification model using Support Vector Classifier (SVC) on the MNIST dataset. This task involves training SVC models with various kernels, including linear, polynomial, and radial basis function (RBF), as well as performing hyperparameter tuning using *GridSearchCV* and *RandomizedSearchCV* for the parameters C and gamma.

Dataset:

The MNIST dataset comprises 70,000 28x28 grayscale images of handwritten digits (0-9). The dataset is divided into 60,000 training images and 10,000 testing images.

1. **Data Preparation: (2 points)**
    - Load the MNIST dataset using `from keras.datasets import mnist.`
    - Flatten each input image into a vector of length 784.
    - Normalize the pixel values of the images by dividing them by 255.
    - Select the first 10,000 samples for training and the first 2,000 samples for testing.

2. **Model Training with Different Kernels (3 points):**
    - Train SVC models with the following kernels: linear, polynomial, and RBF (Keep all other parameters at default values).
    - Evaluate each model's performance on the testing set.
    - Generate classification reports including precision, recall, F1-score, and accuracy for each kernel.

3. **Hyperparameter Tuning (6 points):**
    - Perform hyperparameter tuning for C and gamma parameters using both GridSearchCV and *RandomizedSearchCV*.
    - For *GridSearchCV*, search over the grid: 'C': [0.1, 1, 10, 100] and 'gamma': [1, 0.1, 0.01, 0.001].
    - For *RandomizedSearchCV*, explore the same parameter space as *GridSearchCV.*
    - Identify the best hyperparameters for *C* and *gamma* from both methods.

4. **Model Training with Best Hyperparameters (3 points):**
    - Train the SVC model with RBF kernel using the best hyperparameters obtained from the tuning step (*GridSearchCV)*.
    - Evaluate the model's performance on the testing set.
    - Report the classification report, including precision, recall, F1-score, and accuracy.

5. **Visualization (1 points):**

    - Plot the confusion matrix for the final SVC model trained with RBF kernel.

Support Vector Regression

Objective:

The objective of this assignment is to conduct a regression analysis on the California Housing dataset using Support Vector Regression (SVR). The task involves splitting the dataset into training and testing sets, training an SVR model with default parameters, evaluating the model's performance in terms of mean squared error (MSE), and performing hyperparameter tuning using GridSearchCV to improve model performance.

Dataset:

The California Housing dataset contains information about housing prices in California. It includes features such as location, median income, housing median age, total rooms, total bedrooms, population, households, and median house value. The target variable is the median house value.

1. **Data Preparation:**

    - Import the California Housing dataset using `fetch_california_housing` from `sklearn.datasets`.
    - Split the dataset into training and testing sets with a test size of 20% and a random state of 30.

2. **Model Training with Default Parameters (3 points):**

    - Train an SVR model with default parameters where epsilon(ε) is set to 0.5.
    - Make predictions on the testing set.
    - Calculate the mean squared error (MSE) between the predicted and actual target values and print the MSE.
    - Create a scatter plot visualization of the predictions versus the ground truth.

3. **Hyperparameter Tuning using GridSearchCV (3 points):**

    - Define a range of epsilon(ε) values from 0 to 2.5 with a step size of 0.1.
    - Set up a parameter grid for GridSearchCV with the epsilon(ε) values.
    - Perform 10-fold cross-validated grid search using GridSearchCV to find the best epsilon(ε) parameter.
    - Print the best epsilon(ε) parameter obtained from GridSearchCV.

4. **Model Training with Best Hyperparameter (3 point):**

    - Train a new SVR model with the best epsilon(ε) parameter obtained from the previous step.
    - Predict the target variable for the testing set.
    - Calculate the mean squared error (MSE) between the predicted and actual target values and print the MSE.
    - Create a scatter plot visualization of the predictions versus the ground truth.

Submission:

A .zip file containing the python source code and a PDF report file. The final name should follow the template: _.zip. For example, if your roll no is 15CE30021, the filename for Assignment 6 will be: Assign-6_15ce30021.zip

1. Include an IPython Notebook (.ipynb) file containing the code implementations and outputs of your experiments.

   This notebook should provide a clear and structured explanation of your models, experiments, and results. **(1 point)**

2. Submit a single Python code file (.py) containing the implementations of your models and experiments. The first two lines of this file should include your name and roll number for identification purposes.

Note: both .py and .ipynb files need to be submitted to receive credit on the assignment.