Lab 4 - SVM Classification

Lab 4 on SVM Classification for DS3010 - Machine Learning

OVERVIEW & PURPOSE

In this lab, you will experiment with the SVM classifier.

Instructions

1. Please submit the assignment through Moodle in .ipynb format (python notebook)
2. The submission should contain a single notebook containing all the solutions, including the requested documentation, observations, and findings.
3. The naming convention for the notebook is <firstname>_<lastname>_<rollnumber>.ipynb
4. You must adequately comment on the code to improve its readability.
5. The lab is worth 5 points
6. This graded lab is due on September 29th at 5.00 pm

Lab

SVM Classifier

1. **Linear Data Classification (1.5 points)**
    a. Read through the documentation of the SVM class in the SKLearn and describe the parameters - *C, kernel, gamma, and degree*.
    b. Load the train and test CSV file with the names data_linear_train.csv and data_linear_test.csv.
    c. Plot the data points in a 2D plot with different colors for the two classes.
    d. Create an instance of the SVC(kernel=” linear”, C=0.1) and fit the model.
    e. Plot the SVM boundary regions learned by the classifier.
    f. Print the classification report for training data.

g. Predict the labels y for the data points provided in data_linear_test.csv and store them in a new column named ’linear_kernel_prediction’ in this CSV file.

h. Submit the updated test CSV file.

2. **Non-Linear Data Classification (1.5 points)**

   a. Load the train and test CSV file with the names data_nonlinear_train.csv and data_nonlinear_test.csv.

   b. Plot the data points in a 2D plot with different colors for the two classes.

   c. Create an instance of the SVC(C=0.1) for different kernels(“**linear**”, ***“poly”, “rbf”***) and fit the model.

   d. Plot the SVM boundary regions learned by each kernel classifier.

   e. Print the classification report for training data.

   f. Predict the labels y for the data points provided in data_nonlinear_test.csv and store them in this CSV file's new column named ’nonlinear_[kernel]_prediction’.

   g. Submit the updated test CSV file

3. **Observations (0.5 point)**

   a. Write your observations with comparisons for the above two questions.

4. **Hyper-Parameter Tuning (1.5 points)**

   a. Load the CSV file (wine_fraud.csv) and use the train_test_split function to create the train and test splits.

   b. Do the pre-processing on the dataset.

   c. Define a param_grid dictionary with the list of permissible values for the hyper-parameters **“C”,” kernel”, “gamma”, and “degree”**.

      i. Kernel - [‘**kerne**l’, ***‘poly’, ‘rbf’]***

      ii. Vary C between **1e-5 to 1 in multiples of 10**.

      iii. Gamma - [‘**scale**’,’ **auto**’]

      iv. Degree - [**2,3,4**]

   d. Select the best hyperparameter value to train the final model.

   e. Print the classification report for test data.