LAHORE UNIVERSITY OF MANAGEMENT SCIENCES
Syed Babar Ali School of Science and Engineering

EE212 Mathematical Foundations for Machine Learning and Data Science
Fall Semester 2022

Programming Assignment 4 – Applications of Supervised Learning

---

Total Marks: 100

Submission: 23:55, Sunday, December 11, 2022.

Goal

The goal of this laboratory is to learn different techniques used in supervised learning. The category of supervised learning that we will be dealing with in this lab is called a classification problem. We will train and test our classifier on Diabetes and Bordeaux wine dataset.

Instructions

Name your files Task1.py, Task2.py, and so on. Compress them in a **single** file and name it as LabXX_YourRollNumber. Submit this file on LMS before the deadline. Late submissions will not be accepted.

Before starting, import the following libraries from python:

```
import numpy as np
from matplotlib.image import imread
import pandas as pd
import matplotlib.pyplot as plt
import math
from sklearn import svm
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn import metrics
from skimage.feature import hog
from sklearn.decomposition import PCA
```

In case you are using a Google Colab jupyter notebook, import these libraries:

```
from google.colab import files
import io
```

Task 1: Support Vector Machine (35 marks)

Raw Data

The given dataset is a record of different age group people who are either diabetic (D) or nondiabetic (N) for their blood glucose level reading with superficial body features like body temperature, heart rate, blood pressure, etc.

Classifier

In supervised learning, the data is split into two parts; training data and testing data. We use the training data along with its labels, to train our model. For testing, we only input the testing data and not the labels. We use our trained model to predict the labels of this testing data and compare it with the actual labels (D or N) to see how accurate our model is.

The classifier that we will be using is a binary class Support Vector Machine (SVM). Without getting into much detail, SVM tries to learn the boundary between different classes. You can imagine data as a scatter plot, with data points of each class lying together as a cluster in a 2D plane. Now imagine drawing a boundary such a boundary separates/isolates each class from each other in the plane. SVM tries to learn that boundary and uses it to classify data.

Guidelines

You are required to implement the binary classifier in python whose algorithmic design constitutes the following steps:

1. Download the diabetes dataset file uploaded on the LMS and save in your local directory.

2. Read the dataset and convert it into a pandas dataframe.

3. Now you have the predictors and the target class labels. Split the data into training and testing sets using an **sklearn** library. You may try with a 70/30 or 80/20 test size.

4. Fit your SVM classification model on the training set and evaluate the model performance on the test set using the test set.

5. Once you have the predicted labels, you can make a confusion matrix. It is a matrix with predicted labels on one axis and actual labels on the other. The number in the cell indicates the number of times the corresponding actual label was classified as the corresponding predicted label. The diagonal of course represents the correct predictions while the off-diagonal terms are wrong predictions.

6. Using the in-built function, ‘metrics’, find the confusion matrix of actual and predicted labels of the testing data.

7. Using the confusion matrix, calculate the False Positive Rate (FPR) for each class and the overall accuracy of the model.

Task:2 SVM with PCA

Raw Data

The data that we will be using in this lab is Bordeaux wine data. This dataset contains a total of 14,349 wine reviews collected to understand the relationship between wine quality and its characteristics. The purpose of this task is to predict the wine quality (score in this case) by classifying it into high (≥ 90), medium (≥ 75 and ≤ 89), and low (≤ 74).

Part: A (35 marks)

In this task, you are required to construct a multi-class classifier in python (using SVM without PCA) by following these steps:

1. Download the Bordeaux wine dataset file uploaded on the LMS and save it in your local directory.
2. Read the dataset and convert it into a pandas dataframe.
3. You may omit (at max) two variables from the dataset provided to you. Justify your variable (feature) omission in the case of an SVM classifier.
4. Replace the score variable with the class labels (high, medium, and low) based on the specified criteria.
5. Now you have the predictors and the target class labels. Split the data into training and testing sets using an **sklearn** library. You may try with a 70/30 or 80/20 test size.
6. Fit your SVM classification model on the training set and test the model performance on the test set using the test set.
7. Once you have the predicted labels, you can make a confusion matrix. It is a matrix with predicted labels on one axis and actual labels on the other. The number in the cell indicates the number of times the corresponding actual label was classified as the corresponding predicted label. The diagonal of course represents the correct predictions while the off-diagonal terms are wrong predictions.
8. Using the in-built function, 'metrics', find the confusion matrix of actual and predicted labels of the testing data.
9. Using the confusion matrix, calculate the False Positive Rate (FPR) for each class and the overall accuracy of the model.

Part: B (30 marks)

In this task, the Principal Component Analysis (PCA) technique will be used to reduce the dimensionality of the data. As studied in the course, PCA is a linear dimensionality reduction technique that embeds higher dimensionality data into a lower dimensionality subspace. This is enabled by the linear transformation to retain the principal components which account for most of the variation in the original higher dimensional data. In this part, you are required to implement the multi-class classifier using SVM with PCA.

1. The implementation steps are similar to the steps in part A, except for the extraction of principal components.

2. Before training your binary classifier, extract the principle components of data and then project your original data along those components. The variable *n-components* determines how many components you wish to consider. In this case, check for different *n-components* values (2, 5, 15, 40) and observe the dimensions of the training set.

3. Scale the data and use SVM to predict labels (high, medium, and low) as you did in part A.

4. Compute the confusion matrix along with the accuracy and FPR for each class. Your accuracy might be less than the one you got earlier in part A while using raw data. This is because perhaps the number of components you are using are not enough.

5. Comment on the time taken for classification using PCA as compared to using raw data. Why do you think there is a difference? Why does increasing the number of components led to improved accuracy?