[Project Code: DPSVM1]

Diabetes Prediction using Support Vector Machines

Project Duration : 25-Feb-2024 ~~ 16-Mar-2024
Submission Information : (via) CSE-Moodle

Objective:

A medical lab has decided to build a machine learning model for predicting diabetes for females with age more than 21 years. The model will take the various diagnostic results (like glucose level, blood pressure etc.) and patients' conditions (like age, BMI, pedigree function) as input features and predict whether they have diabetes. Moreover, they have decided to use SVM with squared hinge loss as the machine learning model. Your task is to help the company to build the model.

In SVM with the squared hinge loss, the following optimization problem is solved:

$$\min_{\mathbf{w}\in R^d, b\in R} \quad ||\mathbf{w}||_1 + \frac{C}{2} l(y_i(\mathbf{w}^T\phi(\mathbf{x}_i) - b))$$

where $l(x) = (\max(0, 1-x))^2$ is the squared hinge loss, $||\mathbf{w}||_1$ is the l_1 penalty and

$$\{(\mathbf{x}_0, y_0), (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n-1}, y_{n-1})\}$$

is the training set with the labels y_i s being either +1 or -1. Note that you can check the sign of $\mathbf{w}^T\phi(\mathbf{x}_i) + b$ to determine the predicted class. If it is positive, then the predicted class is 1 and -1 otherwise.

Your tasks are the followings:

1. First, you will devise a stochastic gradient descent algorithm to solve the above optimization problem as follows:
    a. Derive the gradients of the objective function of the above optimization problem with respect to the parameters w and b.
    b. Then, select a suitable termination condition for the stochastic gradient descent iterations.
    c. Also select a proper learning rate schedule for the stochastic gradient descent algorithm.
    d. Finally, devise the full stochastic gradient descent iterations.
2. Secondly, you will write a class to implement the classifier with the following methods:
    a. Constructor: It will take the necessary hyper-parameters like C and initial learning rate (and others if any) and initialize the module.
    b. Train: It will take the train data as input and learn the values of the parameters w and b. Note that the training requires solving the above optimization problem using the devised stochastic gradient descent algorithm.
    c. Predict: It will take the test data as input and return the predicted labels on the data.

    You cannot use any library like sklearn.linear_model.**LogisticRegression** in this part.

3. Finally, you should generate results on the given data and compare its results with the sklearn module sklearn.linear_model.**LogisticRegression.**

Relevant information:

Dataset Filename: `diabetes.csv`
Number of Classes: 2
Data Description:
Number of Instances: 768
Number of Attributes: 8 (all numeric)
Attribute Information:

1. Pregnancies
2. Glucose
3. BloodPressure
4. SkinThickness
5. Insulin
6. BMI
7. DiabetesPedigreeFunction
8. Age

Tasks to be done:

1. **Train-test split:** The dataset is not divided into train and test sets. First randomly split the data into train-test split with 80-20 ratio. You will be using only the train split for your training. The test split will be used only for the final evaluation. Even for the hyper-parameter tuning, you cannot use the test split.

2. **Data pre-processing**: Normalize each feature of the dataset to have zero mean and unit variance. Note that while normalizing the features, their mean and variance should be computed over the train split only. Once, the mean and variance is computed using only the train split, you normalize the test split using the mean and variance computed over the train split.

3. **Implementation of the model:**
   a. Implement the SVM model as stated in the Objective Statement.
   b. Train the model using the train split of the dataset. Note that the training also involves the hyper-parameter tuning. Thus, for hyper-parameter tuning, you have to either split the train split into train and validation again or use the cross-validation on the train split. Whichever method you follow for the hyper-parameter tuning, clearly mention that in your report.
   c. Evaluate your trained model (with the best hyper-parameters) on the test split. And compare the results with the sklearn module sklearn.svm.**LinearSVC.**

4. **Outcomes and Reporting:** Prepare and submit a report with the following –
   a. The devised stochastic gradient descent algorithm.
   b. Results of the hyper-parameters tuning.
   c. Results on the test split of dataset.
   d. You need to calculate precision, recall, f1-score and accuracy for all the experiments.

Submission Details: (to be submitted under the specified entry in CSE-Moodle)

1. ZIPPED Code Distribution in CSE-Moodle
2. A brief (2-3 page) report/manual of your work
   (with your hyperparameter tuning results also presented in that report)

Submission Guidelines:

1. You may use one of the following languages: C/C++/Java/Python/Matlab.
2. Your Programs should run on a Linux Environment.
3. You are **not** allowed to use any library apart from these (Also explore all these libraries if doing in Python, or equivalent of these):

```
import numpy # linear algebra
import csv # data processing, CSV file I/O
import pandas # data processing, CSV file I/O
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
import sklearn.svm.LinearSVC
import operator
from math import log
from collections import Counter
```

   Your program should be standalone and should **not** use any *special purpose* library of Machine Learning for the SVM classifier (apart from the library required to solve the dual optimization problem). Numpy and Pandas may be used. And, you can use libraries for other purposes, such as generation and formatting of data.
4. You should submit the program file and README file and **not** the output/input file.
5. You should name your file as <GroupNo_ProjectCode.extension>.
   (e.g., *Group99_DPSVM1.zip* for code-distribution and *Group99_DPSVM1.pdf* for report)
6. The submitted program file *should* have the following header comments:
   # Group Number
   # Roll Numbers : Names of members (listed line wise)
   # Project Number
   # Project Title
7. Submit through CSE-MOODLE only.
   Link to our Course page: https://moodlecse.iitkgp.ac.in/moodle/course/view.php?id=561

You should not use any code available on the Web. Submissions found to be plagiarized or having used ML libraries (except for parts where specifically allowed) will be awarded zero marks.

For any questions about the assignment, contact the following TA:
Suvadeep Hajra (Email: suvadeep.hajra@gmail.com)