Scikit Learn Lab 2

Spring 2018

Due Date: March 28, 2018

Instructions

- This lab will involve coding in SciKit Learn. You are free to build on the examples presented in the class or develop your own code.
- All instructions for compiling and running your code must be placed in the README file. Please write brief reports for both parts.
- All work submitted must be your own. Do not copy from online sources. If you use any references, please list them.
- You should use a cover sheet, which can be downloaded from here
- You are allowed to work in pairs i.e. a group of two students is allowed. Please write the names of the group members on the cover page.
- **You have a total of 4 free late days for the entire semester. You can use at most 2 days for any one assignment. After four days have been used up, there will be a penalty of 10% for each late day. The submission for this assignment will be closed 2 days after the due date.**
- Please ask all questions on Piazza, not via email.

1 Parameter Tuning of Models on the Digits Dataset [20 pts]

In class, we worked with the Digits dataset that is available as part of Scikit datasets without any parameter tuning. In this part, you will use Grid search for automatic parameter tuning of the algorithms that we have learned so far in the course.

1.1 Algorithms To Train

You have to train the following algorithms on the Digits dataset. The parameters are also mentioned. It's up to you to choose the number of values for each parameter, but you should convince us that you have put in the effort to find the best set of parameters. If you don't understand the meaning of a parameter, please lookup the documentation page. Links are provided with the name of the algorithm in the list below:

List of Algorithms and Prameters

1. Decision Tree
   Parameters: At least four out of the following:
   max_depth,
   min_samples_split,
   min_samples_leaf,
   min_weight_fraction_leaf,
   max_features,
   max_leaf_nodes,
   min_impurity_decrease

2. Neural Net
   Parameters: At least four out of the following:
   hidden_layer_sizes,
   activation,
   alpha,
   learning_rate,
   max_iter,
   tol,
   momentum,
   early_stopping

3. Support Vector Machine
   Parameters: At least four out of the following:
   C (Error Penalty),
   kernel and associated parameters i.e. if you choose polynomial kernel, you need to vary *degree* or if you choose rbf kernel, you need to vary *gamma*
   max_iter,
   random_state

4. Gaussian Naive Bayes
   Parameters Only one parameter available:
   priors

5. Logistic Regression
   Parameters: At least four out of the following:
   penalty,
   tol,

C (inverse of regularization strength),
fit_intercept,
class_weight,
max_iter,
multi_class

6. k-Nearest Neighbors
   Parameters: At least four out of the following:
   n_neighbors,
   weights,
   algorithm,
   p (power parameter for the Minkowski metric)

1.2 Evaluation Methodology and Metrics

For evaluating these algorithms, you first need an evaluation methodology, which will determine how data is split into train/test parts. You can use either of the following:
- random train/test split for a minimum of 5 times OR
- k-fold cross validation with value of k greater than or equal to 5

For evaluation metrics, you need to output as many of the following as possible:
- confusion matrix for all classes
- accuracy measure
- classification report
- area under ROC curve
- ROC plot

1.3 What to Submit

Submit the following:
- Code
- Summary of output including the best parameters for each algorithm
- A brief report in plaintext format analyzing the reports. Which method(s) performed best and why do you think so? What could be done to improve the results?

2 Working with Text Data [20 pts]

Text data is the most common form of data and is widely used in machine learning. In this section, you will learn techniques for pre-processing and model building using text data.
First of all, you will need to work through some examples and become familiar with text processing techniques. Below is the link to a tutorial on SciKit Learn:
http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

You can read more about the multinomial naive Bayes (MNB) class on this link You can read more about the algorithm here

2.1 Applying MNB to a Real Dataset

You will now apply your knowledge of text processing to a real world dataset. You will perform steps like tokenizing, word frequency calculation, etc and then train a MNB model. You are free to use either a random train/test split or cross-validation. You will report evaluation metrics such as accuracy, precision, recall, etc.

You can select either one of the following datasets:
- Twitter US Airline Sentiment dataset. It contains text data and labels, which could be one of the following: negative, neutral, or positive.
- Sentiment Labelled Sentences Data Set. It has 3 datasets from IMDB, Amazon, and Yelp, with review text being classified as either positive or negative. You have to use any one out of these 3 datasets.

2.2 What to Submit

Submit the following:
- Code
- Summary of output including the best parameters MNB algorithm
- A brief report in plaintext format analyzing the results. Are you satisfied with the results? What do you think could be done to make the results better?