Assignment 2: Scikit-learn

Kazi Injamamul Haque

December 2018

1 Introduction

The end outcome of this assignment is to get familiarize with the most used machine learning library used in python, Scikit-learn, along with some other open libraries for data manipulation and visualization such as, numpy, scipy and matplotlib. For this assignment, among the thee given datasets, the spambase dataset was used to learn support vector machine (SVM) classifier and then testing the classifier on the test dataset. This report compares the performance between learned classifier without cross validation and with cross validation by evaluating accuracy, precision, recall and F1-score. In the next section, the methodology is explained along with a short description of SVM and the data that has been used. The results obtained and a short discussion of the obtained results is in the Results and Discussion section. Further, the report is concluded with remarks based on the obtained results in the last section of the report.

2 Methodology

Among the three given datasets, ”spambase” dataset was chosen for this assignment. The data was already split into train and test datasets with separate csv files for data and target (label). The label is either the example at hand is a spam or not a spam, therefore this classification problem is a binary classification problem. Furthermore, the data contains 57 predictors with all numeric values. The train data contains 3680 examples while the test data contains 919 examples. Since it a tall dataset (more examples than predictors), training the model with support vector machine with radial basis function kernel was selected. The baseline accuracy provided with the assignment document is 0.63152 and the outcome of the assignment is to surpass this baseline accuracy.

Firstly, the training of the SVM classifier was done without any cross validation with hyperparameters C=10 and gamma = 0.02. The kernel that was used for training the model was radial basis function(rbf). After training the classifier, the test data was used to produce a report for the different evaluation metrics.

Secondly, the training of the SVM classifier was done with cross validation on different values for gamma. For this dataset, 4 fold training was carried out with

4 different gamma values such as, 0.1, 0.05, 0.02 and 0.01 while keeping track of the best gamma value that produced the best accuracy scoring on training dataset. Finally, using the best gamma that produced the best accuracy score, the final classifier was trained. The test data was used afterwards to produce a report of different evaluation metrics in order to evaluate how the classification model performs. The obtained results and analysis is in the next section of the report.

In addition to the above mentioned steps, a learning curve is also computed and shown in the next section that depicts how accuracy score gets better with increasing number of examples in the training data.

As an extra module, principal component analysis(PCA) was exploited to understand the intuition behind the data provided. In short, PCA transforms a higher dimensional data into lower dimensional data so that we can easily visualize the labels in a 2-D scatter plot using the two principal components (after transformation, the principal components that explain the most variance) that describe the data. Figure 1 Visualizes the data in lower dimension using PCA.

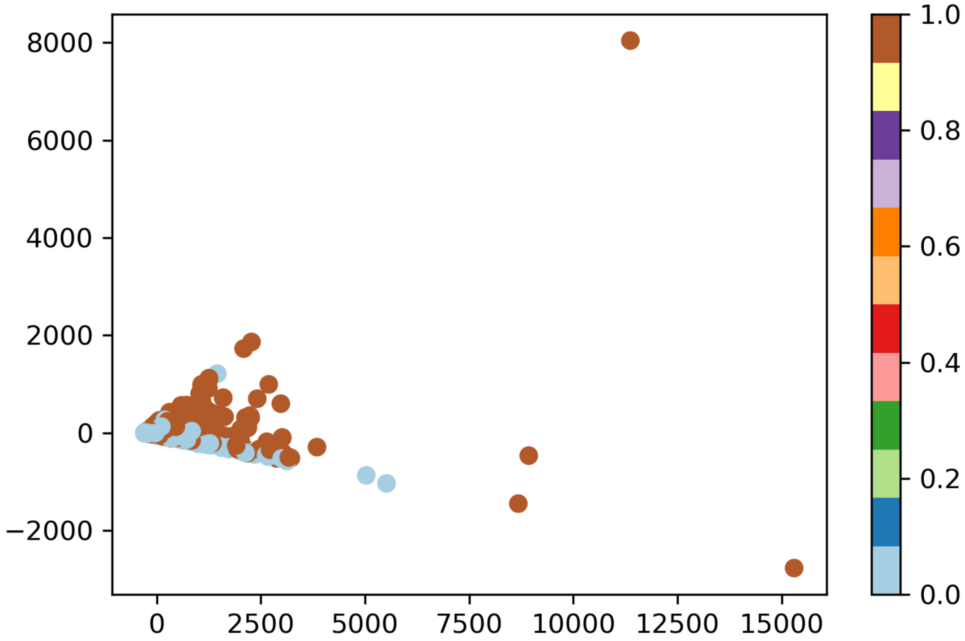

Figure 1: Visualize the data with PCA

3 Results And Discussion

3.1 Without Cross Validation

SVM classifier with rbf kernel function with hyper-parameters C=10 and gamma = 0.01 produces an accuracy score of 84%, which is already better than the baseline accuracy that was provided. Table 1 shows the detailed report on precision, recall and f1-score of the prediction of two classes along with the accuracy score.

Label	precision	recall	f1-score	support
0(!spam)	0.92	0.84	0.87	580
1(spam)	0.76	0.87	0.81	339

Accuracy	0.848748639826
Gamma	0.02
C	10

Table 1: Test Report: without cross validation

3.2 With Cross Validation

Even though the baseline accuracy was crossed without the cross validation process, we would like to achieve better accuracy by tweaking the hyper-parameters and keeping track of for which value it produces the best accuracy score. We can check for best values for both C and gamma, but in this case, cross validation was done for gamma values only {0.1, 0.05, 0.02, 0.01}. Since we checked for four gamma values and try to extract the best gamma value among those, it is a four-fold cross validation process for training the classifier. Table 2 depicts the average score values for accuracy, precision, recall and f1-score for each gamma value it takes to train the model.

Gamma	Accuracy	Precision	Recall	F1-score
0.1	0.752446428089	0.848810722173	0.464795187445	0.600122381339
0.05	0.81168608431	0.741603148047	0.813878415617	0.775634478114
0.02	0.832069402731	0.765843283569	0.837328089272	0.799607552912
0.01	0.855166050434	0.809078114155	0.835950302774	0.822068910093

Table 2: Avg. Accuracy, Precision, Recall, F1-Score over cross validation folds for gamma values

We can see that for gamma value 0.01, it produces the best mean accuracy on the training set among the four folds. Therefore, we take the best gamma value (0.01) to train the final classifier model to be used to evaluate with the test data. After predicting the classes using the test set, we obtain accuracy of 86% on the test dataset. Table 3 shows the detailed report of different evaluation metrics score along with the accuracy score and the respective hyper-parameters.

Label	precision	recall	f1-score	support
0(!spam)	0.92	0.87	0.89	580
1(spam)	0.79	0.86	0.83	339

Accuracy	0.866158868335
Gamma	0.01
C	10

Table 3: Test Report: with cross validation

3.3 Learning Curve

Learning curve refers to a plot of the prediction accuracy or error vs. the training set size. That means, how better does the classifier model get at predicting the target in a test data as number of instances or examples used increases to train the model. In this case, Figure 2 shows how better the model gets in the sense of accuracy score as the number of training examples increases.

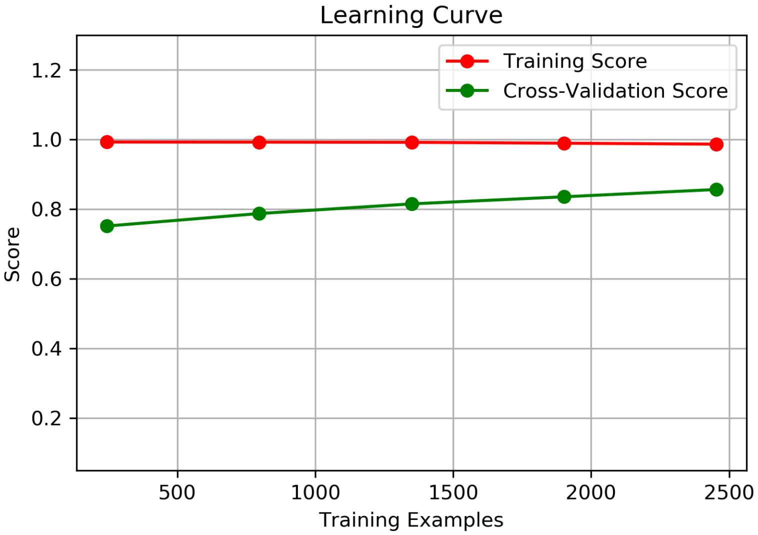

Figure 2: Learning Curve

4 Conclusion

Table 4 compares the results of the two SVM classifiers with different gamma values. In conclusion, we can comment that to use SVM classifier to train models on any kind of data, cross validation process is very useful to achieve higher accuracy. Since we do not know the best values for the hyper-parameters that will give us the best result, it is always recommended that we cross validate models with different values of the hyper-parameters and take the best ones to train our final model. In our case, we cross validated for gamma value and found out after four fold cross validation that the best gamma value is 0.02 which yields the accuracy score of 0.866 on the test data.

Classifier	Accuracy	Gamma
SVM(w/o KFold)	0.848748639826	0.02
SVM(with KFold)	0.866158868335	0.01

Table 4: Comparison of SVM Classifiers

5 Online References

1. https://en.wikipedia.org/wiki/Support_vector_machine
2. https://scikit-learn.org/stable/auto_examples/model_selection/plot_learning_curve.html
3. https://disi.unitn.it/ passerini/teaching/2018-2019/MachineLearning/
4. https://en.wikipedia.org/wiki/Principal_component_analysis