Analysis Report of the Classification Exercise-5

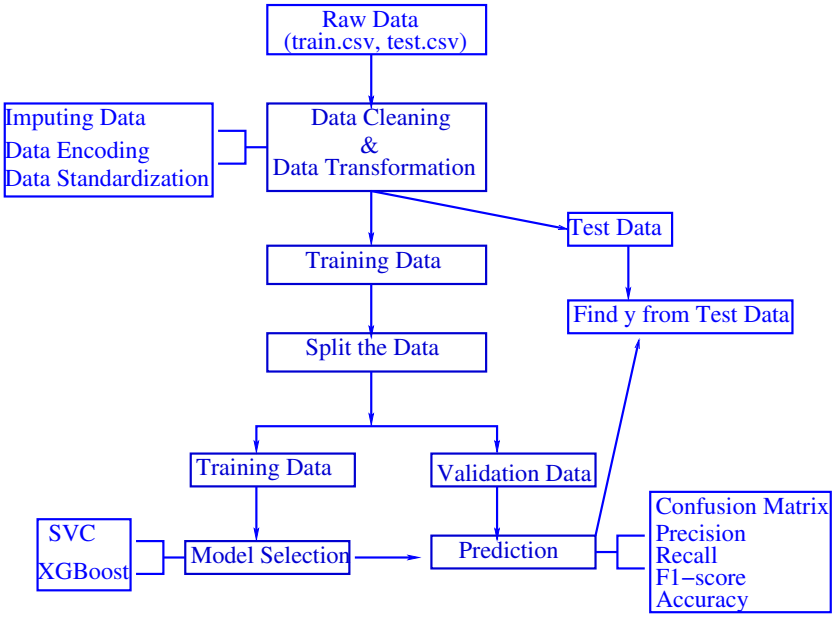
Ramesh Subedi

08-26-2019

Introduction

The flow diagram shown below depicts an analysis scheme. The following two machine learning algorithms were used to classify the label (y variable) of the given data, and make predictions for the unknown test data (unknown to the training data):

1. Extreme Gradient Boosting classifier (XGBoost), and
2. Support Vector Classifier (SVC).



The data preparation steps of imputing missing values, data encoding, and standardizing the features, as described below, were applied to both training data and testing data. Out of 100 features, 51 of them were dropped by using RFE (recursive feature elimination) module of *sklearn*. Two code-files in python were submitted: one file (stateFarmEx5_modelTuning) for model tuning and feature elimination, and another file (stateFarmEx5_Final) for model prediction.

Data Cleaning

There were one hundred features (x variables) and one label (y variable) in training dataset. The test dataset had the same one hundred features as in training data, but no label. The label did not have any missing values. Almost all features, both in training and test dataset, had missing values. The features were imputed by column-median value for numeric variables and the column-mode value for categorical variables.

There were six categorical features, with names $(x_{34}, x_{35}, x_{41}, x_{45}, x_{68}, x_{93})$, and the rest of the features were numeric in both datasets. Feature x_{34} was for car makers like Toyota, Honda, BMW, etc., x_{34} was for weekdays, x_{41} was representing some dollar amount (once the dollar sign was stripped, the variable was converted to a float type), x_{45} was representing a percentage value (once the percentage symbol was stripped, the veriable was then converted to a float type), x_{68} was for months of a year, and x_{93} was representing America, Europe, and Asia. The categorical variables $(x_{34}, x_{35}, x_{68}, x_{93})$ were converted to float type by using *LabelEncoder* method of *preprocessing* module, since the machine learning models required only numerical features (not categorical ones).

One of the other requirements for using machine learning models was to standardize the variables with a mean of zero and standard deviation of one. This was achieved by scaling feature of preprocessing module of *sklearn*.

Extreme Gradient Boosting (XGBoost) Classifier

XGBoost is an implementation of gradient boosting decision trees designed for speed and performance. It internally has parameters for cross-validation, regularization, tree parameters, etc. Boosting is a sequential technique that works on the principle of ensemble. It combines a set of weak learners and delivers improved prediction accuracy. At any instant t, the model outcomes are weighed based on the outcomes of previous instant t-1. The outcomes predicted correctly are given a lower weight and the ones miss-classified are weighted higher. A weak learner is one which is slightly better than the random guessing, for example, a decision tree whose predictions are slightly better than 50%.

For the data size at hand where the number of rows were very very high compared to the column number, the XGBoost was very natural classifier to use - it was quick and rigorous. The time for running 40k of data was less than two minutes. The precision and recall were almost as high as that of support vector classifier. The overall accuracy was 97.72% (i.e., 97.72% of validation samples were correctly predicted versus 2.28% of validation samples being mis-identified). The AUC score was 95.26%.

Support Vector Classifier (SVC)

In the support vector classifier algorithm, each data item is plotted as a point in n-dimensional space (n being number of features) with the value of each feature being the value of a particular coordinate. Then the classification is performed by finding a hyperplane that differentiates the two classes very well. SVC can be used in both supervised and unsupervised learnings - this analysis being supervised learning (known label). SVC is very effective when the number of features is very high compared to the number of data samples (rows) - hence it is useful in text classification problems in natural language programming. In the current analysis, even when the number of features was 100 and number of rows was in thousands, SVC was still reasonable, and gave better result. The precision and recall were better than that of XGBoost. The overall accuracy was 99.08%, the AUC score being 98.24%.

SVC versus XGBoost

SVC and XGBoost are two most popular classifiers with totally different algorithms. SVC is an algorithm based on finding the best decision boundary (the maximum margin hyperplane) to segregate the classes (different than decision tree based algorithms) and effective for a high dimensional space (number of features being more than number of rows).

XGBoost uses ensamble of boosting decision trees, and using many such trees, it decides which datapoint belongs to which class. It is fast and easy to use. Though XGBoost is a fast and robust classifier, it sometimes has an overfitting issue if parameters are not tuned properly.

SVC was chosen in this analysis as a better classifier due to it's better AUC score (and better accuracy score) compared to XGBoost. XGBoost was chosen for an alternative model due to it's fast performance and moderately high AUC score. In addition, we can view plots of decision trees (shown in code file stateFarmEx5_Final) so that we can visually watch what is happening. Such visualzation is not possible in SVC.

Result

The analysis results were summarized in table below where the left-side of table was from SVC and the righ-side from XGBoost. The 40k training samples were divided into 30% for validation data (12k samples), and 70% for training data (28k samples). Out of 12k validation samples, 9540 samples were identified as class 0, and 2460 samples were identified as class 1. These numbers were shown under the title 'support' in table below.

The AUC score from SVC was 98.24% and that from XGBoost was 95.26%. The overall accuracy from both classification was even higher (99.08% from SVC and 97.72% from XGBoost). The precision and recall scores for class type 1 (not 0) from SVC were very high (99% and 97%, respectively). The same result from XGBoost was 98% for precision and 91% for recall.

SVC					XGBoost				
Confusion Matrix					Confusion Matrix				
	Predicted 0	Predicted 1				Predicted 0	Predicted 1		
True 0	9507	33			True 0	9486	54		
True 1	78	2382			True 1	220	2240		
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.99	1.00	0.99	9540		0.98	0.99	0.99	9540
1	0.99	0.97	0.98	2460		0.98	0.91	0.94	2460
micro avg	0.99	0.99	0.99	12000		0.98	0.98	0.98	12000
macro avg	0.99	0.98	0.99	12000		0.98	0.95	0.96	12000
weighted avg	0.99	0.99	0.99	12000		0.98	0.98	0.98	12000
AUC score:		0.9824					0.9526		
Overall accuracy score:		0.9908					0.9772		

Conclusion

The predicted result from the test data (data from test.csv) using both classification schemes were saved for submission: result1.csv from XGBoost and result2.csv from SVC. Two separate code files (stateFarmEx5_modelTuning and stateFarmEx5_Final) were also submitted.

The Support Vector Classifier was found to be better than Extreme Gradient Boosing classifier as the various scores from SVC were all higher than that from XGBoost. For scores from SVC for class type 1 were: AUC 98.24%, overall accuraccy (99.08%), precision (99%) and recall (97%). Such scores from XGBoost were: AUC 95.26%, overall accuracy 97.72%, precision 98%, and recall 91%.

Appendix

An example of decision tree from XGBoost is shown below.