DATS 6202: Individual Final Report

Zhoudan Xie

1. Introduction

The final project performs supervised machine learning to predict a borrower's default risk on a loan. The data is from a Kaggle competition, provided by a loan company called Home Credit. The dataset covers a large number of loan features such as the loan type (cash or revolving loan), the gender of the client, the number of children the client has, and the annual income of the client. The dataset also has the target—whether the client had payment difficulties for the loan (i.e. default or not default).

We employed three learning algorithms to estimate the relationship between the target and a selected set of key loan features, including the Support Vector Machine (SVM), Random Forest, and Naïve Bayes. In doing this as a team, each of the three team members performed one algorithm, including developing the code, writing the corresponding sections in the final report, and recoding the video presentation on the algorithm. In the project, I was responsible for the SVM, and also conducted most of the data preprocessing work.

2. Description of Individual Work

The original dataset is very large, covering 120 features and 307,000 observations. In the data preprocessing, I deleted all the observations with missing values, encoded all the categorical features, and normalized the numerical ones. In performing the SVM, I selected 15 key loan features. However, since the training dataset is highly imbalanced, with 8,076 observations in class 0 (not default) and only 526 observations in class 1 (default), I resampled the data by randomly deleting the observations in the over-represented class (i.e. class 0). As a result, I created a new balanced training dataset with 526 observations in each class.

I then split the training dataset into Train and Test using the train_test_split function in the sklearn.model_selection module, where the Test data are set to be randomly selected 30% of the training data. The random_state option is set to be 100 to define the seed used by the random number generator.

Next, I performed the SVM training by using the sklearn.svm.svc package. In doing so, I defined a function to allow for tests using various kernels including linear, RBF, polynomial, and sigmoid. I examined and compared the performance of each kernel by looking at the classification report, confusion matrix, and receiver operating characteristic (ROC) curve.

In the group final report, I wrote the sections related to the SVM as well as the conclusion section, including sections 3A, 4A, 5A, and 6. In the video presentation, I recorded the portion related to these sections.

3. Results

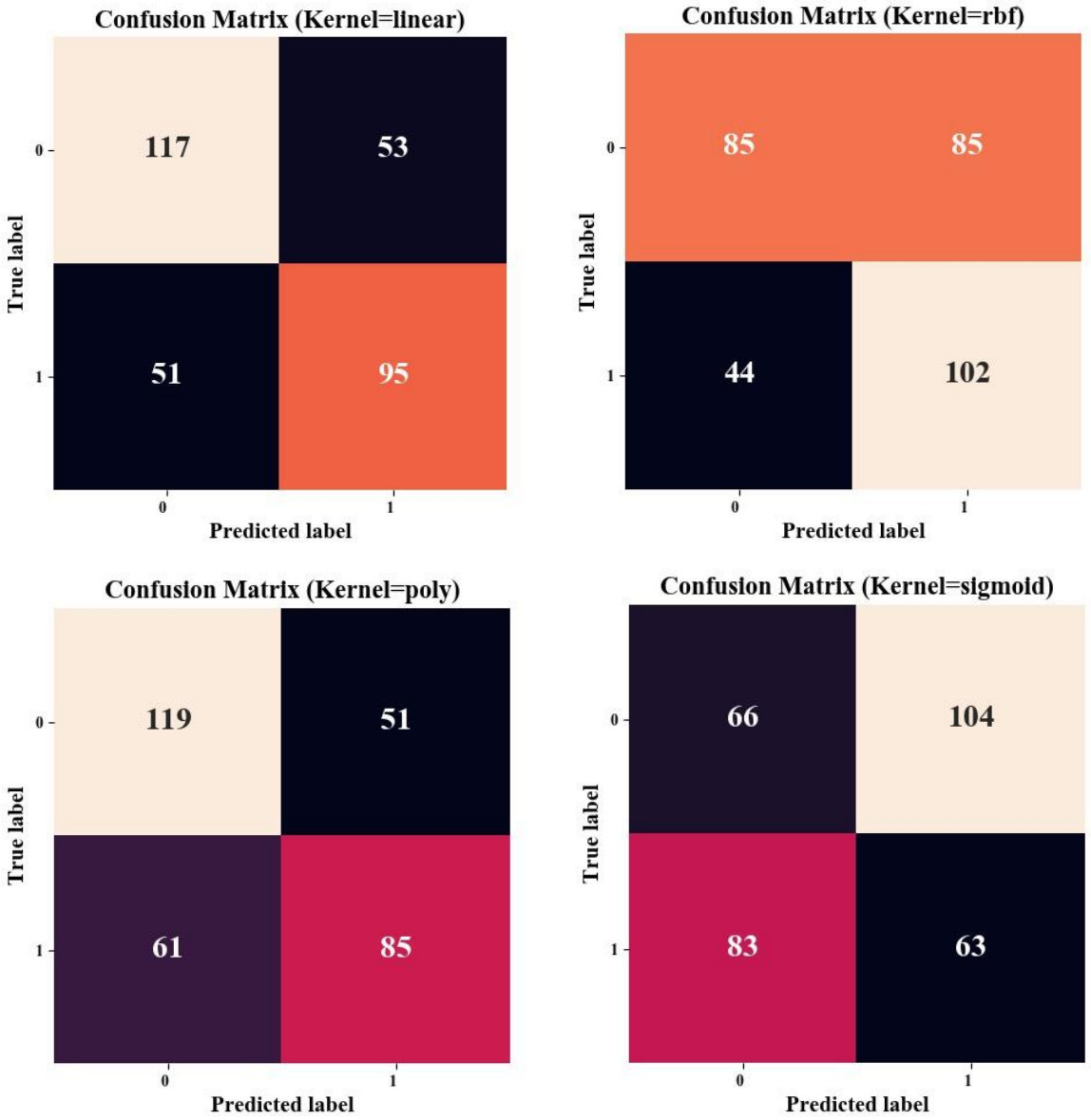
The SVM classifiers using different kernels produce different results. As shown in Table 1, the classification reports shows that the linear and polynomial kernels produce a relatively higher level of accuracy. Their average F1 scores are respectively 0.67 and 0.64. The RBF kernel has a slightly lower score of 0.59, and the sigmoid kernel performs the worst with the lowest score of 0.41.

Table 1: SVM Classification Reports

	Precision	Recall	F1-score	Support
Kernel=linear				
0	0.70	0.69	0.69	170
1	0.64	0.65	0.65	146
Avg/ total	0.67	0.67	0.67	316
Kernel=rbf				
0	0.66	0.50	0.57	170
1	0.55	0.70	0.61	146
Avg/ total	0.61	0.59	0.59	316
Kernel=poly				
0	0.66	0.70	0.68	170
1	0.62	0.58	0.60	146
Avg/ total	0.64	0.65	0.64	316
Kernel=sigmoid				
0	0.44	0.39	0.41	170
1	0.38	0.43	0.40	146
Avg/ total	0.41	0.41	0.41	316

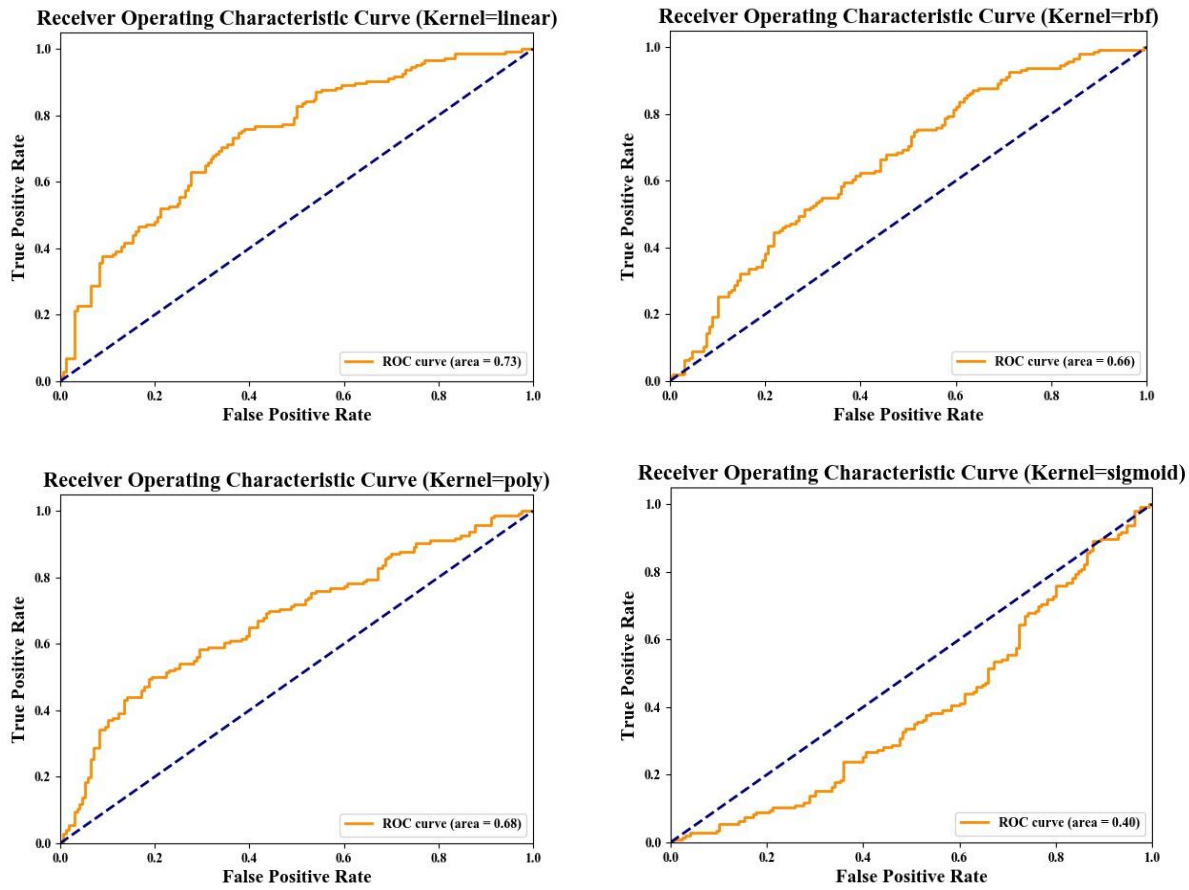
The confusion matrix presents similar results. As shown in Figure 2, the linear SVM produces 117 true positives, 95 true negatives, 53 false positives and 51 false negatives. In other words, the algorithm is able to correctly predict 117 defaults and 95 non-defaults in the test, while incorrectly predicts 53 actual non-defaults as defaults and 51 actual defaults as non-defaults. Comparatively, the polynomial SVM produces a few less false positives but more false negatives. Although the RBF SVM has less false negatives, it generates much more false positives. Still, the sigmoid SVM performs the worst by producing more false positives and false negatives.

Figure 2: SVM Performance - Confusion Matrix



The area under the ROC curve gives us another measure of test performance. As shown in Figure 3, the linear SVM has the largest area under the ROC curve (0.73), which can be considered to be "fair" performance. The polynomial and RBF classifiers present slightly smaller area, 0.68 and 0.66 respectively, and the sigmoid classifier only has an area of 0.40.

Figure 3: SVM Performance - ROC Curves



In sum, the above three performance metrics have conveyed similar messages about the SVM classifiers. In general, the linear and polynomial SVM perform relatively better in terms of accuracy, as indicated by the F1 score and the area under the ROC curve. The RBF has a lower level of accuracy, whereas it might also be a reasonable option to consider for loan companies like Home Credit, because it produces less false negatives and more false positives. Given a similar level of accuracy, lenders may prefer a more "secure" algorithm that produces more false positives than more false negatives. In other words, lender may rather prefer to falsely reject 100 applications than falsely approve one application. From this perspective, the RBF has its advantages over the other kernels. Nevertheless, neither of the SVM classifiers above actually achieve a very good performance, so we test the other algorithms as well.

4. Summary and Conclusions

In this project, we performed supervised machine learning to predict a borrower's default risk on a loan. Using a dataset provided by Home Credit, we employed three learning algorithms, including the Support Vector Machine, Random Forest, and Naïve Bayes. We tested the performance of the three algorithms by examining their accuracy scores, confusion matrix, and

ROC curves. As a result, we identified the superior algorithm that can best predict a borrower's default risk.

In particular, I performed the SVM learning using various kernels including linear, RBF, polynomial, and sigmoid. I found that the linear and polynomial SVM perform relatively better than the other two in terms of accuracy in the test. The linear SVM achieves the highest F1 score and the largest area under the ROC curve, indicating a fair performance. Although with a lower level of accuracy, the RBF SVM may have more practical advantages because it produces less false negatives than the others.

5. Code

See the Individual-Code file for the code I wrote. In writing the code, I copied the code on SVM from Prof. Amir Jafari's Hithub, and made minor edits on them. That is approximately 60 lines. There are 184 lines in my individual code, so the percentage of the code I copied is approximately 30%.