Logistic Regression and SVM

Michael Childs and Saichand Thota

Introduction

For this project, we used two datasets to perform logistic regression and SVM analysis on. First dataset is about loan payback and the second is about Persistency of drug. Through our studies, we calculated optimal models for future predictions of individuals repaying loans and persistence. We utilized logistic regression and SVM to predict the output classed because the goal variable in both sets is categorical and these methods are used to identify when events occur between classes. Our models help in understanding the persistence of the drug (persistence or not) according to the doctor's prescription as it is one of the challenges facing by all pharmaceutical business and to loaning companies.
Github Repository: Here
Presentation Slides: Here

Dataset

Both the datasets were sourced from Kaggle. First dataset was a reduced collection from an original dataset from LendingClub.com and only included individual 2016 approved loan results in a quarterly as well as a unified source csv. This csv that was used for this project. There were 9578 loan items to process and predict from, and overall a very hearty dataset. It necessitated minimal preprocessing (aside from a few minor adjustments which will be discussed below) thanks to the dataset provider already cleaning percent symbols and converting percentages to floats. Likewise, this source identified the best column features to perform regression on and reduced the original 120+ columns down to only 20. Thus, this dataset was perfect to perform analysis on. This dataset had the following features to use: credit policy (according to LendingClub.com, the *original* source for the dataset, is if a customer meets various underwriting criteria from LendingClub.com), purpose (general reason for the loan), interest rate (of the loan), installment (monthly installments due from borrower), log annual income (natural log of annual income of borrower), debt to income (ratio of borrower), fico score (of borrower), days with credit line (of borrower), revolving line balance, revolving line utilization rate, inquiries in the last 6 months (from creditor TO borrower), delinquencies of borrower over past 2 years, public record (infractions on borrower's public record), and whether or not the loan has been fully paid.

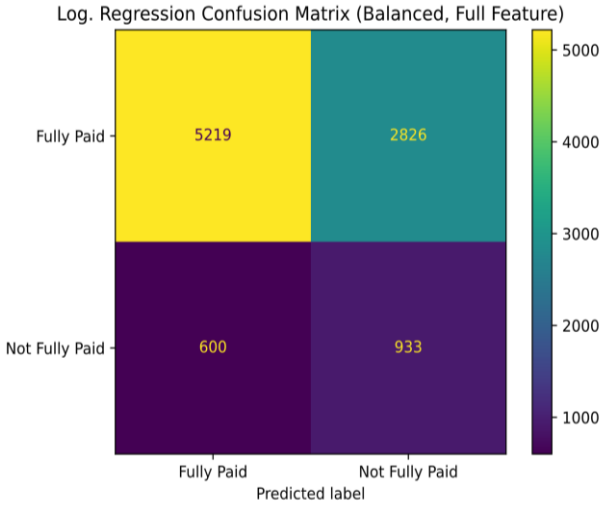
To comprehend the persistence of pharmaceutical medications, a second dataset was used. It has 69 attributes with 3425 patients' details. A few of the attributes, include the patient's basic information, such as age, race, ethnicity, gender, and region, as well as Patient ID, a unique ID for each patient; Persistency Flag, a determination of whether the patient is persistent; NTM Dexa Scan Frequency, the number of DEXA scans taken prior to their first NTM Rx date; and NTM factors like Risk Factors and Injectable Experience.

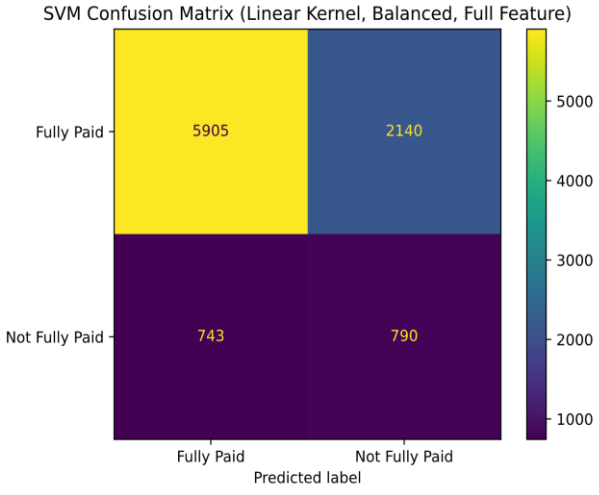
Analysis technique

First Dataset, we used logistic regression and SVM's for our analyses. We used the liblinear solver for the logistic regression model with 3 iterations- one using the "balanced" class weight and all features, one using custom identified class weights and all features, and one using custom class weights and reduced a feature set. For the SVM, all 3 iterations used all features and balanced class weights but varied in using the linear kernel, polynomial kernel, and RBF kernel. When we assessed the loan repayment percentage, we discovered that 0.16% of the population had fully repaid their loans**.** The "purpose" column of the dataset was categorical, and thus needed one hot encoding (using the pandas "dummies" functionality) to easily work with the other quantitative data. All of these columns were used to predict the "not.fully.paid" categorical results, with 1 being fully paid and 0 being not fully paid loan recipients. Finally for the dataset, various pairs of features were compared to identify relations between them. These graphs are provided in the github repo but are not detailed in this report at this time in lieu of more interesting results. **Second Dataset,** prior to moving further with the modelling, we looked for null values and utilized LabelEncoder to convert categorical data into numeric data because most of the attributes were in categorical format. After preprocessing, we implemented the models by splitting the dataset for training (80%) and testing (20%).

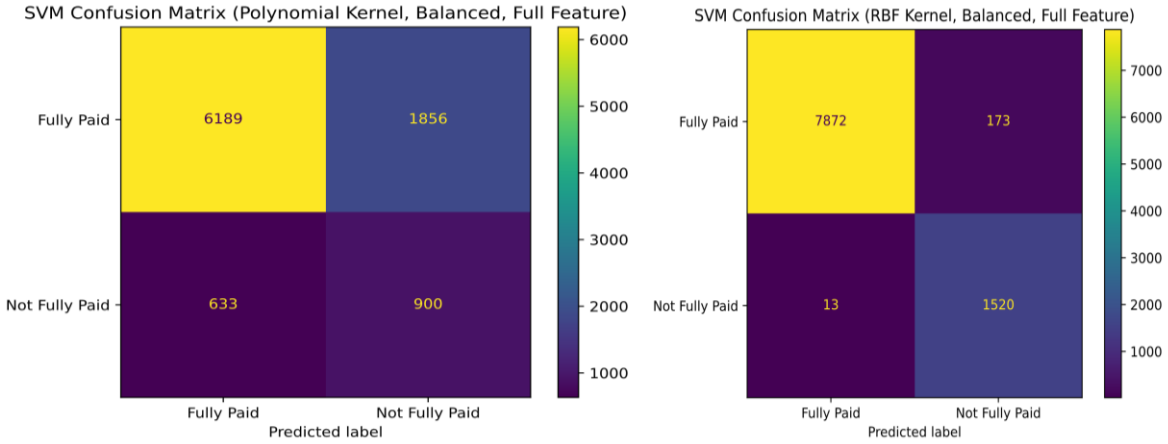
Results

For loan dataset, it was found that the 3 logistic regression models performed relatively the same regardless of the parameters (precision: 0.248, recall: 0.608, F1 score: 0.352). Overall, logistic regression didn't perform very well in predicting whether a loan would be paid off in full or not. It produced the following confusion matrix:





In using SVM, however, the results were much more satisfactory, especially in using the RBF kernel. For the linear kernel (with balanced class weight and full feature set), there was the following confusion matrix and results (*precision: 0.2696, recall: 0.515, F1 score: 0.354):*

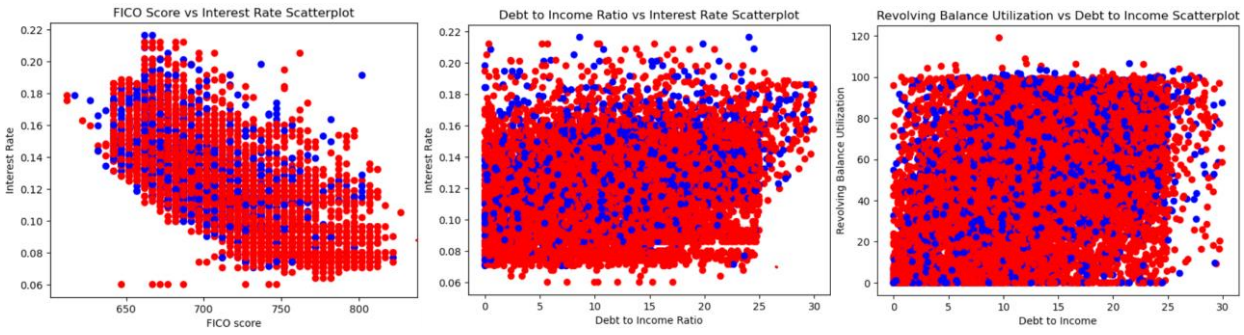


As can be seen, the results are generally the same with the linear kernel. For the polynomial kernel, a degree of 3 was used with an auto gamma (and the same other settings) for the following confusion matrix and results (*precision: 0.327, recall: 0.587, F1 score: 0.419*)

Here, we can see this kernel performs quite significantly better at predicting whether or not an individual has paid off their loan. However, the truly significant results were derived from the RBF kernel with a balanced class weight, full feature set, and gamma of 1, which produced the following results (*precision: 0.898, recall: 0.992, F1 score*: *0.942*):

As is visible, these results are incredibly succinct at predicting an individual's probability of paying off a loan and could certainly be extremely useful to a creditor in deciding on whether or not to approve a loan. The only drawback to this method is that the SVM's and this RBF SVM kernel in particular takes significantly more time to run the model (the logistic regression took roughly 0.1 seconds, the RBF SVM took 9.0 seconds). This is even *after* the dataset was standardized explicitly for SVM usage. Without standardizing, the dataset is simply too large to even run SVM. As such, it certainly has an exponentially increased runtime and necessitates some preprocessing for standardization. However, these results proved to be well worth it. Additionally, varying the class weights outside of balanced or even removing some features only resulted in poorer results, and thus are not recommended or included in this report.

According to the repayment of full loan, we plotted scatter plots between different features.



For Persistence of Drug data, firstly, we determined the uniqueness of attributes to know their type. Only two properties have numerical values, and the rest are categorical, as we discovered. We tested for null values and the count of persistent (1290) and non-persistent (2136) in the dataset. We eliminated the target variable as well as Race, Region, Ethincity, Idn Indicator, and Ptid since we believed they were ineffective for predicting the outcomes (Patient ID). But once they were gone, precision dropped. Thus, we trained all of the features into our models. Below table shows the best models' parameters and their precision, recall, f score, accuracy and run time. We didn't tabulate every model's details because of space constraints, they can be found in code.

Model with Parameters	Precision	Recall	F score	Accuracy	Run time
LogisticRegression(class_weight={0: 0.16, 1: 0.84}, solver='newton-cg')	**0.928**	**0.597**	**0.756**	**0.717**	**0.144**
svm.SVC(kernel='poly', class_weight='balanced', gamma='auto', degree=3,decision_function_shape='ovo')	0.800	0.912	0.852	0.801	0.265
svm.SVC(kernel='rbf', class_weight='balanced', gamma=.1)	0.799	0.820	0.809	0.8	0.552
svm.SVC(kernel='linear', class_weight='balanced', gamma=10)	**0.856**	**0.802**	**0.829**	**0.791**	**1.30**

Although the precision of the SVM's polynomial and rbf are practically identical, the polynomial outperformed the rbf in terms of recall and f score. Using GridSerachCV, we experimented with a wide range of SVM and Logistic Regression parameters to identify the models with the highest results. We discovered that the optimum score for Logistic Regression is superior to SVM's based on the above table and GridSearchCV.

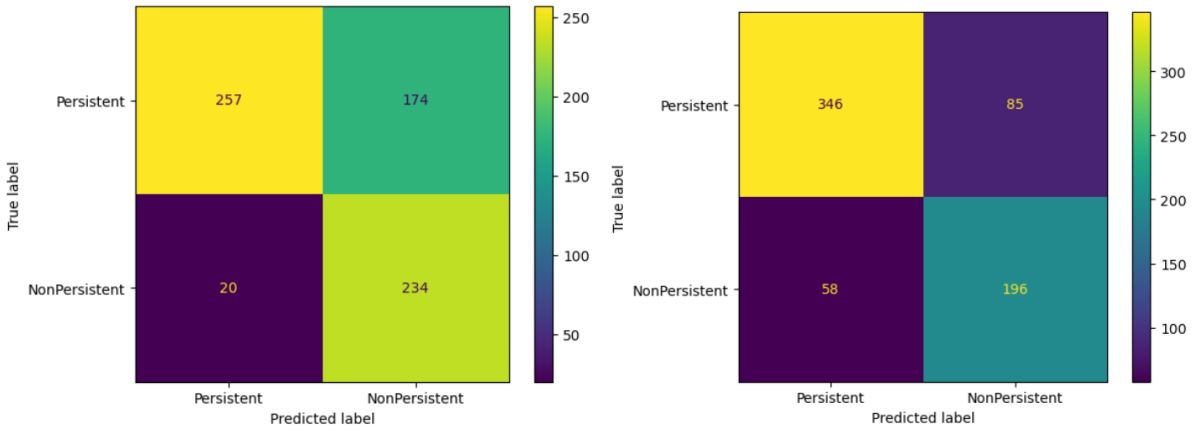
SVM -

```
Best hyperparameters:{'C': 1, 'class_weight': 'balanced', 'decision_functi
on_shape': 'ovo', 'degree': 2, 'gamma': 'auto', 'kernel': 'rbf'}
Best score:  0.8068629151710056
```

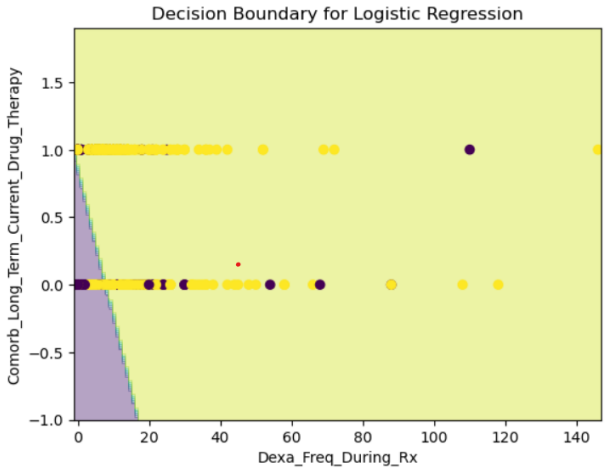
Logistic Regression –

```
Best hyperparameters:{'C': 0.1, 'multi_class': 'auto', 'penalty': 'l1', 's
olver': 'saga'}
Best score:  0.8163553023125477
```

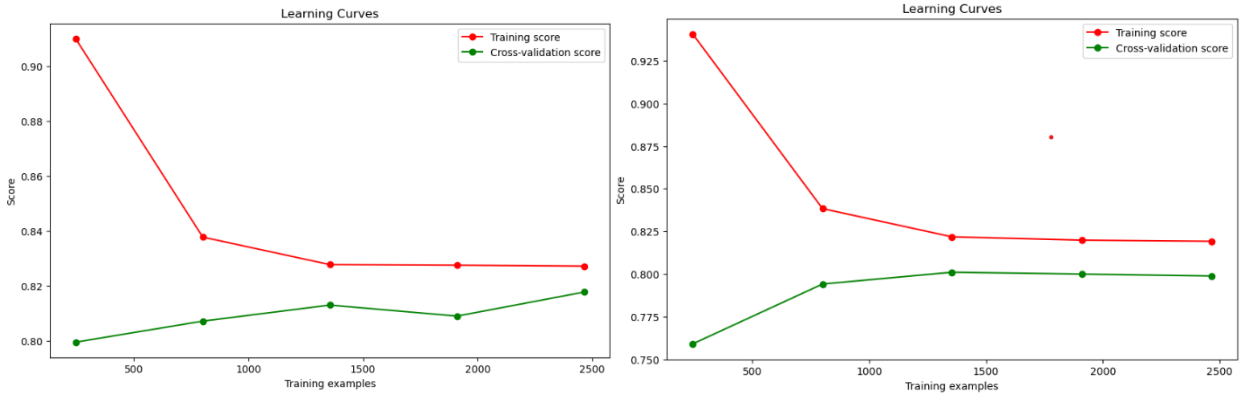
The confusion matrix for the models that did the best is shown below - Logistic Regression and SVM, respectively.



While this dataset has more features than the first, we used GradientBoostingClassifier to determine the key features. Dex Freq During Rx was shown to be more significant than any other variable (0.500), followed by Comorb Long Term Current Drug Therapy (0.086). The top ten features were highlighted in the code. We used above two features as the only features for a Logistic Regression model and plotted Decision Boundary.



To evaluate the generalizability of different models, we used cross validation (10 – fold) and learning curves techniques and our models performed better as size of training data increases.



We didn't spend much time on data preparation or technological difficulties because most of the data is organized and lacks null values or special characters. Most of the code's run time is used by GridSearchCV but not by Logistic Regression or SVM models. Total code run time – 40 min