DATA 401 Project 3 Report Template

Data Preparation

3.1 Data selection

The initial dataset had 122 variables and we began by dropping all features with more than 1000 missing values. After that, we just dropped any remaining rows that contained missing values. Then we dropped the *CODE_GENDER* variable as we felt it could impart historical discrimination into our model. That left us with 59 variables. Finally, we used our own domain knowledge to narrow the 59 variables to just 17 that we thought could be correlated with the target variable of whether or not a person defaults. This included things such as their income or the loan amount. This left us with 17 predictors plus the target column.

After one-hot encoding the categorical variables to numeric variables, the 17 columns turned into 38 columns. To reduce any extra dimensionality that didn't actually provide much useful information to the model, we used Logistic Regression with L1 penalty. L1 regularization shrinks many of the coefficients to zero, which would be the variables we get rid of. This dropped the 38 columns to 16 columns, which ended up being the final columns for our training data. In order to help the model choose the correct columns to drop, we used hyperparameter optimization to choose the L1 penalty that resulted in the best f1 score, and used the coefficients from that model to drop columns. Additionally, sub-sampling was used in model training to balance out the effect of the unbalanced target labels on the model. 10,000 observations for each class label were randomly selected for training.

3.2 Data cleaning

We initially started off with 307511 observations, and we ended up with 306562 after dealing with missing values in the manner explained above. Since we still retained so many observations, we felt that this method of dealing with missing values was acceptable.

We one hot encoded 7 variables that were categorical to make them numeric so that the model could use them. 3 of these variables were binary and remained as one column, while all the others ended up creating more columns. After this step, there were now 36 total columns in the dataframe.

We also modified the variable *AMT_GOODS_PRICE* to be *AMT_GOODS_PRICE / AMT_CREDIT. AMT_GOODS_PRICE* represents the price of whatever the loan is being taken out for and *AMT_CREDIT* represents the amount of credit for the loan. Because of this, the variables were about 99% correlated and so provide almost identical information to the model.

We created the new variable to provide “new information” to the model and to reduce the increase in variance of the prediction due to multicollinearity.

3.5 Data formatting

We standard scaled all of the variables in order to make sure that every variable had the same effect on the model. Also, the standard scaling was important for the gradient descent to converge properly and efficiently.

Additionally, we encoded the target variable to be (-1,1) whenever we used Perceptron and SVM, and (0,1) whenever we used Logistic Regression, as Logistic Regression classifies predictions based on probabilities.

Data Modeling

4.1 Modeling technique

Logistic Regression with L1 penalty was used to reduce dimensionality of the data. This was done on the entire dataset. Additionally, since the target classes were heavily unbalanced, we took a balanced subsample of the data in model training to make sure that coefficients that placed all observations on one side of the boundary were not chosen.

Then we performed Bayesian hyperparameter optimization using Optuna. Besides optimizing on lambda, we also optimized on the class weight as it turned out to be important for predictive accuracy. This was done just using logistic regression. We didn’t use SVM for either of the dimensionality reductions because SVM is by far the slowest.

Since we needed to implement the models ourselves and our implementations didn’t have class weights, we used balanced target label sub-sampling to make the final prediction.

We used Support Vector Machines, Logistic Regression with L2 regularization, and the Perceptron with L2 Regularization to classify the data. Because the goal of the models is to maximize predictive accuracy, we chose to ignore assumptions for linear regression for the Logistic regression and the Perceptron.

4.2 Test design

Since our goal was to compare the three classification algorithms, we needed to do hyperparameter optimization to choose a single L2 penalty strength to use for all three models.

We used cross-validation with Stratified K-folds to validate and tune the models. Stratified means that each fold has the same proportion of target labels. Our optimization score was the f1 score, which we were maximizing.

Something to note is that there were about 91% class 0 labels. Our goal was to maximize the accuracy for predicting class 1 labels, since you could get 91% accuracy just by predicting all class 0. So our overall accuracy would decrease as we increased the f1 score.

Two techniques for dealing with unbalanced labels were class weights, and training on sub-samples. We used class weights for prototyping with sklearn, and sub-sampling for our own final implementations.

4.3 Model creation

Our hyperparameter for the L2 penalty strength (lambda) was chosen using Bayesian hyperparameter optimization. The same hyperparameter was used for all three models. We also optimized on the class weight hyperparameter, which helps deal with the unbalanced data.

For class weights, our final value for lambda was 0.02775159018992187 and our final value for the class weight was 7.839529551742979. Since we couldn't implement a class weight in our own implementations, a separate value of lambda of 0.10819565 was used for our own implementations. That value was also obtained through hyperparameter optimization

Models produced

Logistic Regression	$\hat{\mathbf{y}} = -0.038-0.034x_1+0.012x_2+0.008x_3-0.028x_4+0.417x_5-0.702x_6-0.027x_7 -0.013x_8+0.005x_9-0.191x_{10}+0.016x_{11}-0.086x_{12}+0.075x_{13}+0.030x_{14}-0.019x_{15} +0.100x_{16}$
SVM	$\hat{\mathbf{y}} = 0.020-0.029x_1+0.018x_2+0.002x_3-0.025x_4+0.294x_5-0.371x_6-0.011x_7 -0.006x_8+0.005x_9-0.109x_{10}+0.019x_{11}-0.027x_{12}+0.055x_{13}+0.023x_{14}-0.014x_{15} +0.061x_{16}$
Perceptron	$\hat{\mathbf{y}} = 0.000-0.001x_1-0.003x_2+0.003x_3-0.003x_4+0.116x_5-0.606x_6-0.007x_7 -0.004x_8-0.002x_9-0.096x_{10}-0.002x_{11}-0.056x_{12}+0.011x_{13}+0.007x_{14}-0.004x_{15} +0.028x_{16}$

Features (numbers correspond to the x's):

1) NAME_INCOME_TYPE_Pensioner
2) NAME_INCOME_TYPE_State servant
3) NAME_INCOME_TYPE_Unemployed
4) NAME_INCOME_TYPE_Working

5) NAME_EDUCATION_TYPE_Higher education
6) NAME_EDUCATION_TYPE_Secondary / secondary special
7) NAME_FAMILY_STATUS_Married
8) NAME_HOUSING_TYPE_House / apartment
9) NAME_HOUSING_TYPE_Rented apartment
10) NAME_CONTRACT_TYPE
11) FLAG_OWN_CAR
12) AMT_ANNUITY
13) AMT_GOODS_PRICE
14) EXT_SOURCE_2
15) REGION_RATING_CLIENT_W_CITY
16) DAYS_BIRTH

The following models for each classification algorithm were trained on 20,000 samples, each having all 16 features, 10,000 coming from each class of the target, using gradient descent.

4.4 Model assessment

Our models were validated using F1-score with Stratified Cross-Validation with 5 folds. The metric we used was F1 score specifically for the class 1 targets. As the unbalanced class, we want to make sure the predictions for those have both good precision and good recall.

For our final model with class weights, the f1 score was 0.241 and the accuracy was 0.797. This was with Logistic Regression, which gave the highest scores for both.

For our own implementations without class weights the f1 scores and accuracy are below. The perceptron had the highest scores of 0.435 and 0.790 for both f1 and accuracy respectively.

	Logistic Regression	SVM	Perceptron
F1 Score	0.397	0.393	0.435
Accuracy	0.755	0.751	0.790

Evaluation

5.1 Results evaluation

The goal of this project was to build a model that could accurately predict whether a loan applicant would be able to repay their loan. Overall, the accuracy of the model doesn't look too bad, but the f1 score is really bad. It seems that it is just really hard to correctly classify whether

someone would have trouble or not. We would not recommend using this model to make predictions.

5.2 Discussion

We don't believe there was any important factor that was in the original dataset that has totally been overlooked, other than the issues with the unbalanced data. That said, we believe that the subsampling done in training and the stratified folds for the cross validation in the validation adequately accounted for the lack of balance in the original data set.

When taking into consideration the ethics of our model, we removed the gender variable in order to make sure that our model wouldn't impart any gender bias when predicting whether someone would default on their loan or not. This, in our best estimation, of the usable features, was the sole variable that may have caused ethical concern. We also considered the ethical issues that may come up with features involving the client's income, as that may have had an impact on poorer communities, which may bring up some correlation with different demographics, but we ultimately decided that income was too important of a factor to leave out.

One clear bias in the data was the massive imbalance in the target classes. There were a significantly larger number of 0's than 1's in the target variable, indicating many more samples of people who didn't have difficulty paying back their loan. There was also possible bias in our choices, in selecting features that we thought were correlated with the target. Obviously, our domain knowledge is limited, as we are not experts on loans. There very well could have been features left out or unnecessary variables that were kept.

Given that the goal of the project was to maximize predictive accuracy, it isn't absolutely necessary that the model be completely transparent, as we take a bit of a "black box" approach. With that said however, to maintain ethical standards and to ensure the usage of our models are not harmful to certain communities, especially given the possible impacts of the decisions that may be made with these models, we do need to keep a certain level of transparency. This involves ensuring we aren't introducing different biases in our model with the data we use to train the model.

Again, with the focus mainly on predictive accuracy, interpretation was not a priority for this project. But, we must be mindful of the generalizability of the model. We can only generalize the predictive capabilities of these models to populations that are representative of the data we used to train the models. Assuming, the data used for model development came from the same client that these models were meant for, we can prevent misinterpretations by restricting usage of the models to the clients whose data were provided for the project.