DS 6014: Bayesian Machine Learning Final Project

Jordan Machita, Michael Pajewski, and Buckley Dowdle

Project Description: Our dataset contains information on default payments, demographic information, credit history, payment history, and bill statements of credit card clients in Taiwan. The goal of our project is to not just predict whether or not a customer defaults on their credit card payment but also to measure the uncertainty around such predictions by analyzing the posterior probability of default. The data set contains 30,000 distinct credit card clients. The response is a binary classification of whether or not the customer defaulted. The predictors include the amount of given credit, gender, education, marital status, age, six months of history of past payment, six months of past bill statements, and six months of monthly payment. **Figure 1** depicts the distributions of predictor variables as well as correlation between them. On the diagonal, we can see that distributions are quite similar for the selected predictors between default and not default. Additionally, we can see some interesting correlations, such as a fairly distinct boundary between age and limit balance that separates default and not default data points.

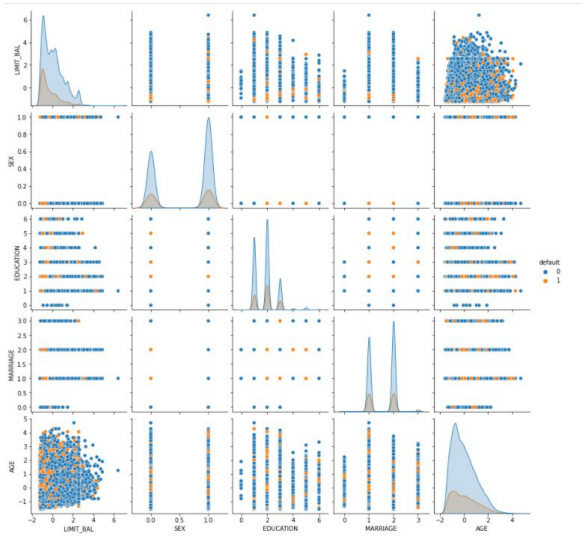

Figure 1: Pair plot showing the distribution and correlation between predictor variables

Mathematical Linkage between the Problem and the Method: By using a Bayesian framework for logistic regression rather than a frequentist approach, we are able to evaluate the uncertainty of not only our predictions but also of each predictor's role in the model. This allows us to identify the impact that changes in the predictor values have on the probability that a customer will default. A key benefit of such an approach is that we can use soft classification for load decision making by evaluating the probabilities and uncertainties instead of making strict predictions based on a predetermined threshold.

Since our response, default, is a binary outcome, logistic regression is an obvious choice. However, by using Bayesian logistic regression, we treat the coefficients of the predictors in the model as random variables that come from some prior distribution rather than fixed values that must be estimated. This Bayesian approach allows us to obtain a mean and variance for each of the distributions of the coefficients, which enables us to measure their uncertainty. We can then use sampling or variational inference to approximate the posterior probabilities.

Bayesian Method Used: For this problem, we implemented automatic differentiation variational inference (ADVI) due to the large size of the data and the need to continually update the algorithm as new data points become available. Variational inference allowed us to examine the posterior distributions of the

model parameters in a computationally efficient and scalable way that sampling could not provide in an acceptable timeframe for this problem. Our implementation of variational inference used expectation maximization to approximate the true distributions by maximizing the evidence lower bound (ELBO), which is equivalent to minimizing Kullback-Leibler divergence. **Figure 2** depicts the maximization process of our model's lower bound over 50,000 iterations, showing clear convergence and indicating it may achieve accurate approximations of the posterior probabilities.

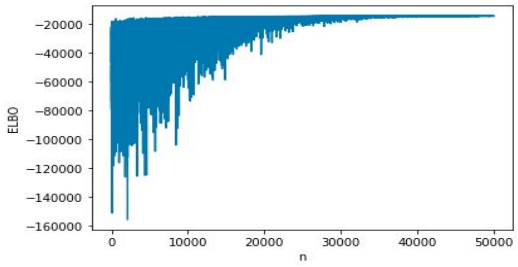

Figure 2: Evidence Lower Bound Maximization of 50,000 iterations

For this model, we used uninformed priors as we had no access to additional data or expert opinion to inform the model. Additionally, the volume of data in this data set would have a negligible effect on the posterior distribution.

Results and Conclusions: We estimate the percentage effect for all the predictors in the model of the 24 predictors; the 10 with the most meaningful percentage effects and or odds ratio are listed in **Figure 3** below. We standardized the payment amount by the individual clients limit balance PAY_AMTX = PAY_AMTX / LIMIT_BAL so payment amounts represent the percent of the customers account limit. While holding all other independent variables constant a one unit increase in payment amount 2 decreases the probability of default 77%. This shows that those who paid a higher percent of their standardized payments amount in the previous month before default are significantly more likely to not default all other predictors held constant. Payment amount 5, bill amount 1 , and limit balance all also significantly decrease the probability of default by 52%, 48%, 27% individually while all other predictors are held constant.

Again in **Figure 3** we can see that a one unit increase in Payment 0 increases the probability of default by 75.9% all other predictors held constant. Payment 0 is the number of monthly payments behind an individual is the month we are predicting if they defaulted or not. The scale of payment 0 is -1 pay on time, 1 payment delay for one month … to 9 payment delay for nine months and above. An increase in payment 0 has the highest effect on the probability of defaulting. This suggests that a person who is farther delayed on their monthly payments is more likely to default on payment.

	mean	sd	hpd_3%	hpd_97%	mcse_mean	mcse_sd	ess_mean	ess_sd	ess_bulk	ess_tail	r_hat	odds_ratio	percentage_effect
Intercept	-1.751	0.027	-1.804	-1.701	0.000	0.000	9213.0	9213.0	9187.0	9312.0	NaN	0.173600	-82.639974
LIMIT_BAL	-0.327	0.019	-0.362	-0.291	0.000	0.000	9490.0	9490.0	9479.0	9833.0	NaN	0.721084	-27.891626
SEX	-0.168	0.029	-0.221	-0.112	0.000	0.000	9910.0	9882.0	9917.0	9495.0	NaN	0.845354	-15.464617
EDUCATION	-0.142	0.013	-0.166	-0.117	0.000	0.000	9468.0	9468.0	9469.0	9755.0	NaN	0.867621	-13.237874
MARRIAGE	-0.258	0.016	-0.287	-0.228	0.000	0.000	10327.0	10327.0	10320.0	9837.0	NaN	0.772595	-22.740477
AGE	0.054	0.016	0.022	0.081	0.000	0.000	9839.0	9839.0	9847.0	9999.0	NaN	1.055485	5.548460
PAY_0	0.565	0.011	0.544	0.584	0.000	0.000	10467.0	10459.0	10476.0	9814.0	NaN	1.759448	75.944778
BILL_AMT1	-0.666	0.042	-0.745	-0.589	0.000	0.000	9691.0	9674.0	9697.0	9695.0	NaN	0.513760	-48.624049
BILL_AMT6	0.279	0.051	0.183	0.370	0.001	0.000	10205.0	10205.0	10209.0	10089.0	NaN	1.321807	32.180734
PAY_AMT2	-1.472	0.228	-1.910	-1.049	0.002	0.002	10395.0	10350.0	10390.0	9921.0	NaN	0.229466	-77.053391
PAY_AMT5	-0.739	0.214	-1.128	-0.338	0.002	0.002	9845.0	9791.0	9845.0	9927.0	NaN	0.477591	-52.240873

Figure 3: Model Summary Table

In the Forest plot, **Figure 4,** we can see Pay amount 6, Pay amount 3, and Bill amount 5 cross 0 and are found not to be significant in the main effects model at $p < 0.05$. These predictors can not help us predict the probability of default. All other predictors are found to be significant. Pay 0, the number of months of payment delay the month of payment default or not default, has the strongest positive influence on the probability of default. Payment amounts 1, 2, and 5 have the strongest negative effect on probability of default but have wide distributions and we are more uncertain of the significance. Bill amount 1 also has a strong negative effect on the probability of default and has a much tighter credible interval than the payment amount predictors.

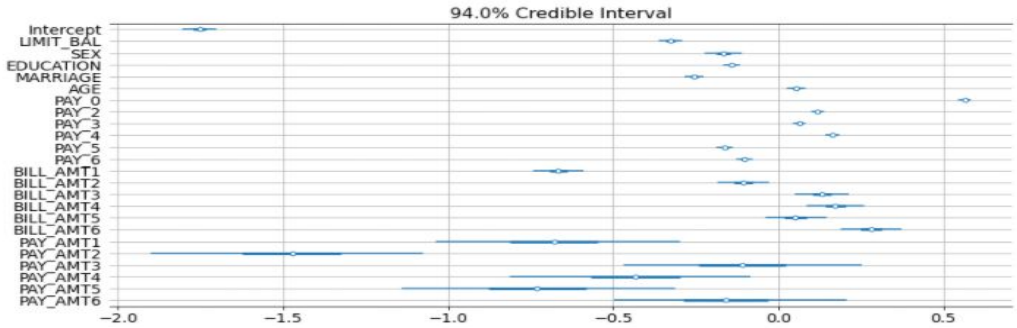

Figure 4: Forest plot indicating the credible interval of the distribution of each distribution for the predictor variables

While the purpose of this project was not to create the most accurate predictor of default, it is worth examining the predictive capability of the model as well as what threshold for classification is most accurate. In **Figure 5**, one can see that the model's AUC score is 0.72 and achieves its maximum F1 score at a threshold of about 0.3. In addition to the information about the model parameter distribution, this is valuable as it allows creditors to adjust this number as their tolerance for risk changes.

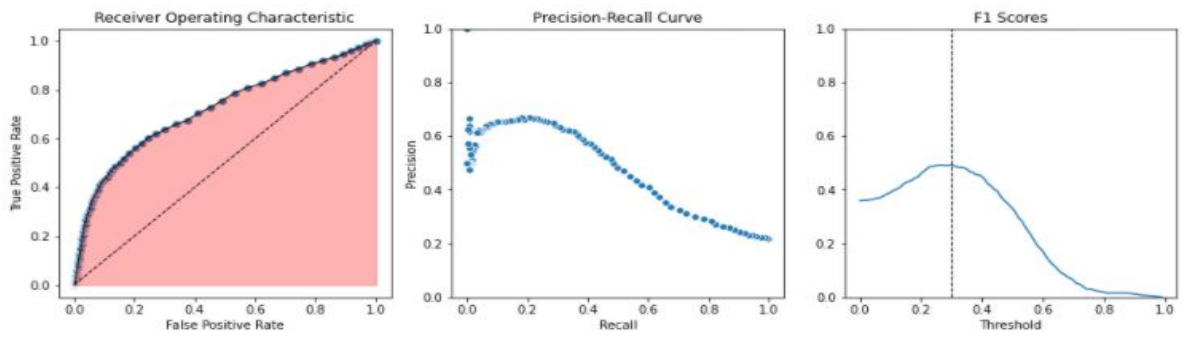

Figure 5: Model performance metrics

One benefit of producing the probabilities of default rather than just the predictions is that we can analyze the distribution of the probabilities. We ran our model on all 30,000 data points to generate the posterior probability of default for each customer and plotted the density in **Figure 6**. The probabilities appear to be fairly normally distributed with a peak around 0.2, but with a large tail on the right. This illustrates that

most of the data points have a low probability of default, but the curve does tend to flatten out into a thicker tail at around 0.3. Such a change in the distribution suggests that 0.3 or 30% would be a good choice for the threshold used for hard classification. We can also see that there are few data points that the model associates with a high probability of default, such as 0.75 or greater. One of the causes of this is that only around 20% of the customers in the dataset defaulted, so the data is somewhat unbalanced. Thus, for hard classification, we should identify the probability value where the normal distribution of not defaulting seems to merge with the peak of what could be a second distribution representing those who default. For our posterior probability distribution, this appears to be around 0.3.

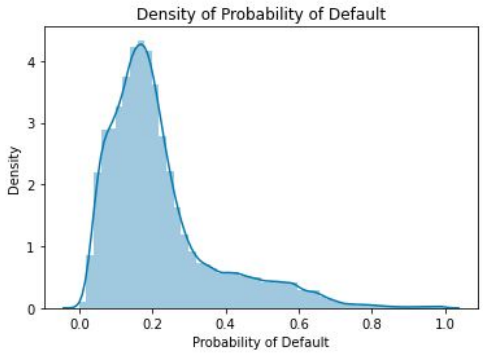

Figure 6: Distribution of posterior probability of default

Using 0.3 or 30% as the threshold we produced binary predictions of credit default. These predictions are represented in the confusion matrix in **Figure 7**. With the predictions we have a Sensitivity of 85.8%, Specificity 53.6%, and Precision 87.9%. The overall accuracy is fairly high but the model was only able to predict default of 53.6% of those who actually defaulted.

Confusion Matrix

	Predicted: Not	Predicted: Default
Acutal: Not Default	20544	2820
Acutal: Default	3376	3260

Figure 7: Confusion Matrix with a threshold of 0.3

By using Bayesian logistic regression, we were able to not only build a model for predicting whether or not customers default but we were also able to obtain the posterior probabilities of default. This allowed us to analyze the uncertainty around our predictions and to understand the confidence of the impact of predictors on default.

References

Li, Susan. (2019, July 22). Building a Bayesian Logistic Regression with Python and PyMC3. https://towardsdatascience.com/building-a-bayesian-logistic-regression-with-python-and-pymc3-4dd463bbb16

Logistic Regression with PyMC3, Goldinlocks. goldinlocks.github.io/Bayesian-logistic-regression-with-pymc3/

Salvatier J., Wiecki T.V., Fonnesbeck C. (2016) Probabilistic programming in Python using PyMC3. PeerJ Computer Science 2:e55 DOI: 10.7717/peerj-cs.55

Yeh, I-Cheng. " Default of Credit Card Clients Data Set." Uci.edu, 2009, archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients