Predicting Defaulting of Credit Card Clients

One of data science's biggest appeals is the ability to analyze and produce meaning by navigating and processing data. Data science provides many methods you could use to approach any problem you encounter in the worksphere via data science. For example, you could gain insights on customers that buy your products, figure out the most optimal method to deal with a situation, or even predict future trends. In this example, we decided to take a look at the "Default of Credit Card Clients Dataset" from Kaggle.com.

Background About the Project

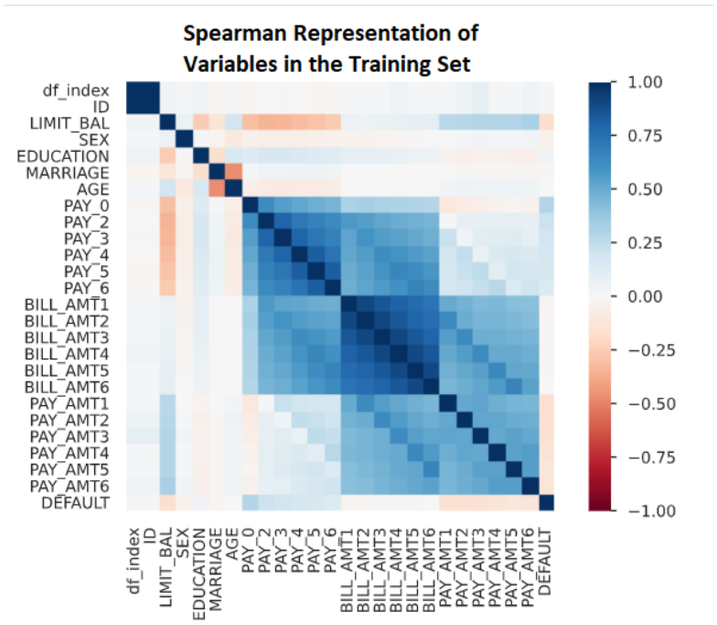
This dataset describes credit card clients in Taiwan from over the course of April 2005 to September 2005. It records the clients' demographic factors, credit data, history of payments, and whether or not they will default on the following payment.

What we are interested in analyzing is if we are able to predict whether a person will default on their next payment based on the clients' other pieces of information. We will build a model that will analyze a person's demographic factors, credit data, and history of payments to predict whether a person will default their payment or not.

Building a Model

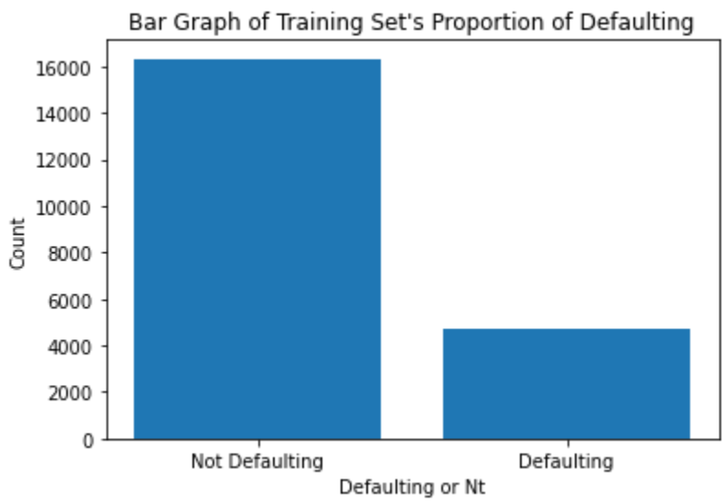
We then start our analysis by splitting the data and quickly inspecting the training set via a spearman representation below. This shows the correlation between each and every variable in the dataset.

Figure 1:



On initial inspection, the most useful values for predicting defaulting are probably a client's limit balance, the most recent pay, and bill amounts. Limit Balance could imply that since they are able to have larger credit values, they may not be able to pay as fast as people with lower limits. For the pay value, the more months a client has spent defaulting the more likely the client would default again. Furthermore, the most recent value would be the most accurate value, because it could potentially lead into the data of the next month leading to defaulting if they were already defaulting before. The same thing for the two previous variables would also apply to the bill value. Higher cost and recency may make a client more likely to default. The ones with the least correlation are the demographic factors. Some examples of demographic factors are gender and marriage status, which make a lot of sense that they don't have much correlation. It is fairly difficult to find a strong argument on why these factors would correlate with defaulting.

Figure 2:



Looking at the proportion of defaulting in the dataset in the above figure, we can assume that we should focus on building a model with a high recall and precision value. Focusing on accuracy would predict non-defaulting too often to get an excellent model.

In an attempt to predict whether a client would default, we built several models using many different approaches ranging from logistic regression, the random forest classifier, and LightGBM. We ended up only focusing on the model that predicted our test set the best after hyperparameter optimization was performed.

Figure 3: Classification Report on the Training Set

	precision	recall	f1-score	support
non-default	0.92	0.83	0.87	16320
default	0.56	0.76	0.64	4680
accuracy			0.81	21000
macro avg	0.74	0.80	0.76	21000
weighted avg	0.84	0.81	0.82	21000

Looking at our results on the training set, we see that we could only maximize our f1-score for defaulting to 0.64 on the training set. This is probably an unsatisfactory result considering that mispredicting could be fairly detrimental to the credit card company.

Figure 4: Classification Report on the Testing Set

	precision	recall	f1-score	support
non-default	0.88	0.80	0.84	7044
default	0.46	0.61	0.53	1956
accuracy			0.76	9000
macro avg	0.67	0.71	0.68	9000
weighted avg	0.79	0.76	0.77	9000

When we test the model on the test set, it results in test scores that are similar to the training scores from before, but a bit lower. This would imply that the results are fairly trustworthy. While we could aim to increase our f1 and recall values, this result still has some level of reliability. There isn't that much optimization bias, as the difference in recall and f1 score between the training and test set is only around 0.1.

Retrospective

In hindsight, while our project may be interesting to look at as a first attempt to approach a problem, it is definitely not a satisfactory model to the problem. While we managed to get a f1 value of 0.53 and a recall value of 0.61, this is probably not a reliable classifier we can use. Mislabeling 40% of positive data as false positives is extremely detrimental, meaning that we should not use this model to detect credit card defaulting.

The model definitely needs improvement if we want it to be trustworthy enough to be used. The parameters we chose during the hyperparameter optimization step may have not given us our desired result. We could have potentially achieved more accurate results if we had chosen different hyperparameters values for the LGBMClassifier in the parameter grid. There is also the choice of using a randomized search instead of a grid search. The models we used could've also been a problem, as they may have not catered to this specific problem well. We could've used a different algorithm such as CatBoost, which would potentially lead to a better score.

One other reason why our results may be inaccurate would be due to feature importances. We should not be overconfident with the feature importances as they are highly dependent on the existence of other features in training the model. One feature may have a high feature importance coefficient solely because it has a strong correlation with an important feature relating to defaulting. As we used almost all the features, we may have used a feature that may not be relevant domain-wise, which could reduce the effectiveness of our model

There were also potential missed opportunities such as feature engineering or feature selection that could've been explored further and maybe improve the result. As we had data for 6 months, we could've modified the data to include some form of time series as a predictor. This could potentially make our features less redundant, and value the more noteworthy predictors more.