Introduction

In this project, we are given a real dataset as well as labels and are asked to solved three different problems:

1. What's the probability of a customer being more than 1 day late on his credit card payment: 162 features per customer
2. What's the probability of a customer being more than 31 days late on his credit card payment: 211 features per customer
3. What's the probability of a customer being more than 61 days late on his credit card payment: 202 features per customer

Methodology

To solve this problem, a multilayered perceptron is used with 5 intermediate layers and ReLU as the activation function at each node. Finally, the sigmoid activation function is used in order to get a probability between 0 and 1.

Libraries Used

- Matplotlib
- Numpy
- SKLearn
- CSV
- Pandas
- Pytorch

Analysis

Unfortunately, the data is very noisy and there is lots of information that is missing. In the MLP model, all inputs are required to be numerical so the data needed a lot of preprocessing. In order to process the data, the following was done:

1. **Dropping columns that are predominantly null**: Some features are null for more than 50% of the datapoints. I felt that assigning values to these fields would do more harm than good as we are using less than half the data to interpolate these missing values. Therefore, these features were simply removed and not considered.
2. **Splitting dates**: Some of the features appeared to signify a specific date, in the specific format YYYYMMDD. Since the data is a date, it didn't make sense to treat it as a continuous variable or a to one hot encode each date on its own. Therefore, I split each of these features into 3 different features, one with the year, one with the month, and another with the day.
3. **Differentiating between continuous and categorical variables**: Some of the features had a very small range of unique values. Thus, they seemed to represent categories more so than continuous variables. Therefore, for all features, if they had less than 30 unique values, the feature is one-hot encoded.

4. **Filling NA values with interpolated values:** Since we didn't get rid entirely of null values in step 1, we still have null values in out data. Thus, we replace the missing values with the average of the column they are in.
5. **Normalizing the data:** Since the data has quite a large range, the features need to be normalized. To normalize all the data, we subtract the minimum of each column and divide by the maximum minus the minimum of each column.
6. **Polishing the data:** Finally, there is a very few columns at the end that are strings that have a lot of unique values. These strings cannot of that much use to the MLP model, so they were also dropped.

The Model

The training model that is used is a multilayer neural network consisting of 5 hidden layers as follows:

- Layer 1 – 512 nodes
- Layer 2 – 256 nodes
- Layer 3 – 128 nodes
- Layer 4 – 64 nodes
- Layer 5 – 32 nodes

All the 5 layers have ReLU as their activation function. For the output layer, sigmoid is the activation function. Finally, the hyperparameters were chosen as follows after intensive testing:

- Number of epochs: 10 – This seemed to be the optimal value after investigation
- Batch size: 128 – Lower and higher values gave slightly worse results
- Loss function – BCELoss, or Binary Cross Entropy as we have a binary classification problem
- Optimizer – Adam
- Learning rate – 0.0001

Results

On the validation set, the AUROC was measured to be the following:

- 1st Problem: 88%
- 2nd Problem: 81.4%
- 3rd Problem: 82.5%

Conclusion

The MLP proved to be a good model since the AUROC is significantly higher than 0.5. To improve these results, one could further try to analyze each feature and perhaps get rid of more features that prove to be uncorrelated or of very low correlation in order to reduce noise.