Springboard Capstone 1 : Data cleaning and wrangling
Lending club loan data
Matthew Gudorf

1. Introduction

The goal of this project is to create two models; the first predicts whether a loan will default, the second predicts the potential recoveries therefrom. The first step for any data science project is to clean and wrangle the dataset, which is what this document details.

Upon importation into a Jupyter notebook, a warning was displayed indicating that the data contained variables of mixed type (e.g. a combination of python lists and floats in one data feature). This is an issue that could be handled by interpreting the imported data as strings, for instance. There are a number of such properties that need to be taken into account before the prediction models can be created. Specifically, my belief is that the important data cleaning decisions to make are: how to handle missing values, redundant information (a feature column identically equal to a single value), multiple types of data. Care must be taken when manipulating the data in these ways, as to not ruin the generalizability of any models.

2. Handling missing values

Firstly, for the missing data values there are a number of options. For instance the missing values can be imputed by a specified method (replace the missing value with the mean of the data feature, replace with a constant value, etc.), the samples containing missing values can be dropped from the data set, or if the feature is a categorical variable, “missing” can become its own category. Imputation is not acceptable as it will ultimately alter the summary statistics of the data, excluding the statistic maintained by the replacement rule. That is to say, if the missing values are replaced with the mean then the mean will be maintained, but other summary statistics will not be maintained. An alternative would be to just drop all samples with missing values; unfortunately the missing values are distributed in such a manner such that if the samples with missing values are removed from the dataset then there are exactly zero samples left to work with.

While the distribution of missing values is not optimal, I developed a strategy to work around them. There are a number of features (columns) wherein the vast majority of values are missing. These are discarded if the percentage of missing values is above a threshold value. When placed in decreasing order, the distribution of these percentages in figure 1 has multiple plateaus. This leads to an interesting choice of where to put the threshold. While it is still arbitrary, the essential choice is whether to include the “plateau” that occurs around 40%. I have

not decided upon the best choice but it is evident that these values correspond to a different time period than the features with a smaller proportion of missing values.

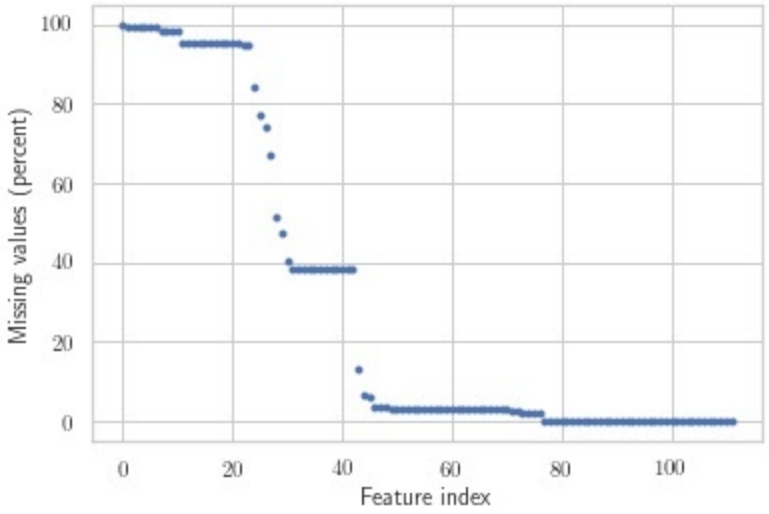

Figure 1 : The percentage of missing values in each feature, in decreasing order.

The main strategy I employ for the current data used in the two-step modeling procedure is to first replace all missing values in categorical feature data with a new category labeled “Missing”. Next, I remove all features with a proportion of missing values greater than 30%. Lastly, the remaining samples that have missing (numerical) values are removed from the dataset, as it is hard to motivate any particular imputation. The main trade-off in this process is that I keep more in exchange for fewer features.

3. Handling pathological features

Once these measures to account for missing values have been performed, I move to the other general measures that are performed to process the data to be ready for each stage of the modeling process. For example, there are some features which contain a single value. As it stands, any partition of the data into training and testing data quite obviously contains only one value. This is unlikely to be an example of sampling bias due to the number of samples; it is simply a systematic issue with how the data is recorded. For example, there is a feature named “policy_code” which only takes a value of 1. The lack of utility stems from the fact that the training domain being a single value has the implication that it won’t know how to handle differing values. There are also features which have a highly imbalanced distribution, namely, the vast majority of samples take one value. This case is not as well motivated as the prior case, but still I believe it is a valid measure to prune these features from the dataset.

One last case of hard to manage data features are categorical features which are idiosyncratic in nature. For example, one such feature would be the description for the reason for the loan provided by the borrower. Excluding certain generic categories such as “debt consolidation”, there are a very large number of unique responses and hence categories that

would need to be encoded. One way of handling this would be to bin all unique responses into an "Other" category. The problem with this approach is that the cutoff would be arbitrary, essential defined by how many categories I deem important. I deemed this too dubious and so I opted for removing these types of categorical features from the dataset, such that they would not be used in the modeling procedure.

4. Dependent variable preparation

The last component of the data wrangling is processing the data to adhere to our modeling hypotheses. For instance, the loan outcome classification portion of this project wants to predict whether or not a loan will be charged off by its maturity date. Therefore, I need to first subset the loans which have matured, and then choose a way of handling the possible outcomes of each loan (recorded in "loan_status"). For the maturity subselection, the data contains information pertaining to loans of three year terms and five year terms, up to December 1st, 2018. Therefore, I took the subset of loans whose issuance date plus term was equal to or less than this date. It could be argued that there should be two models, one for each loan term, as they might represent distinct populations. While I did process the data using this belief, I did not end up creating separate models for the two populations.