Springboard Capstone 1 Statistical Analysis
Lending club loan data
Matthew Gudorf

1. Kolmogorov-Smirnov testing numerical features

The data types in this data set are heterogeneous; even within the subset of features which are defined on a continuous domain (approximately continuous, as most variables monetary in nature), they can have dramatically different distributions. I investigate these distributions, as I believe that these continuous variables will have more of an effect in the modeling process, mainly because I think they are better positioned to capture the individualistic nature of loan borrowers, as they take on more unique values than other variables. My idea is to investigate the numerical features to know how to handle the data in the modeling process, i.e. how to rescale, if a transformation is warranted. The variables I investigate are:

1. The annual income (without some very extreme outliers)
2. Total payments made to date
3. The percentage of credit being utilized
4. Number of open accounts
5. Debt to income ratio

Income inequality is a relatively common topic in pop culture such that it is fairly well known that the distribution of annual income does not follow a normal distribution. It is not clear if this will remain the case for the distribution of annual incomes of loan borrowers. It is questions like these that motivated me to investigate the distributions of the various quantities. Specifically, using the Kolmogorov-Smirnov test and the corresponding p-values I attempt to classify the different distributions that occur in the data set. This could have an effect on my choice for rescaling and engineering of the various features. Because the sample size is so large and I am relatively new to financial data, I would have assumed most features would follow a normal distribution; it turns out I was quite wrong in this regard. The next handful of figures represent histograms for the values of each feature; the red line corresponds to the kernel density estimate produced by the “kdeplot” plot function from the Seaborn package. The y-axis labels on the left correspond to the number of borrowers in each histogram bin, while the right side labels are the density. In this stage, the goal is to find the distribution which matches the histogram so that the sample distribution and hypothesized distribution can be compared via the Kolmogorov-Smirnov test. First off is the distribution of annual income.

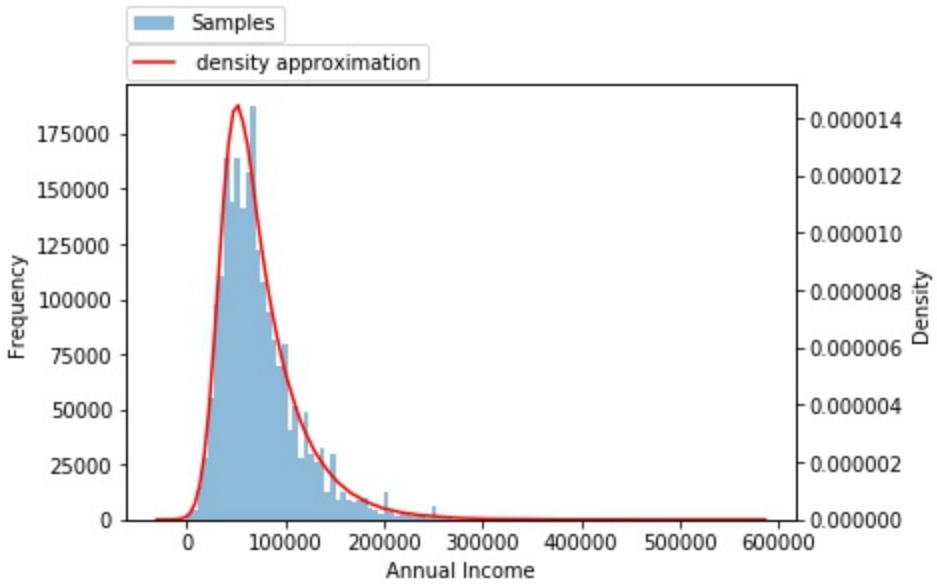

Figure 1. Annual income with exponential-normal kernel density estimate.

My methodology is as follows: look at specific quantities such as skew and kurtosis as well as the crude shape produced by the histogram. For annual income I hypothesized that it followed an exponential normal distribution.

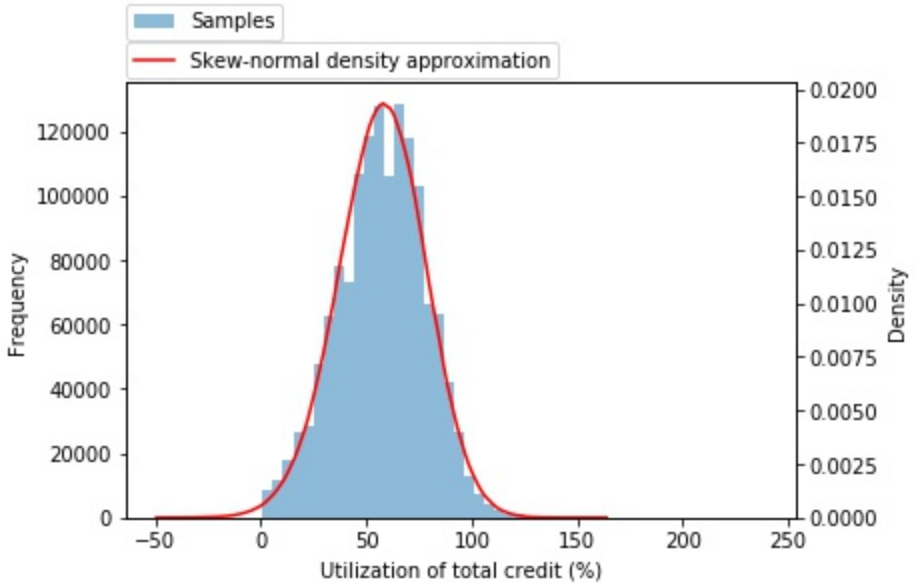

Figure 2. Total credit utilization distribution with skew normal kernel density estimation.

Another quantity previously reported was the utilization of credit; I believed (before visualization) that it followed a normal distribution but it appears upon further analysis that my claim is that it follows a log-normal distribution.

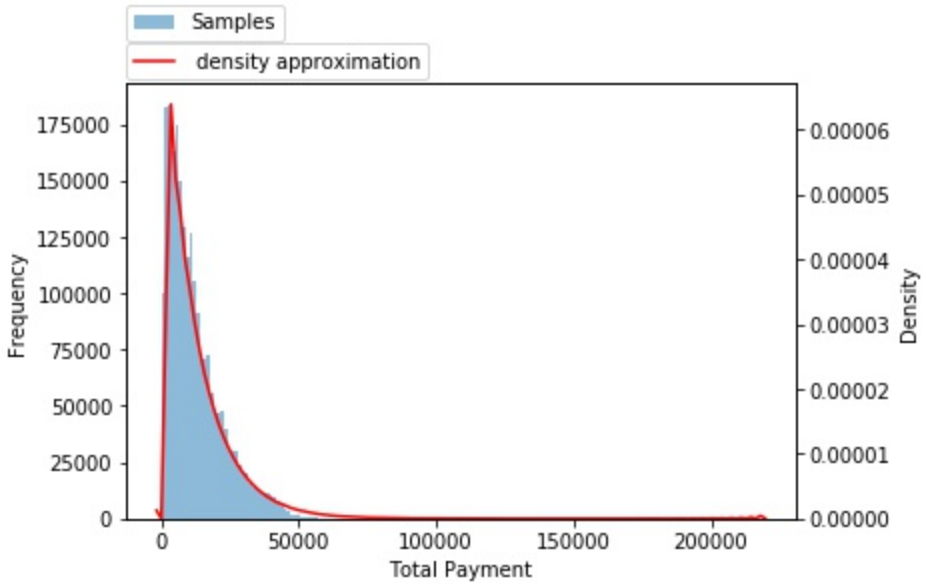

Figure 3. Total payment with exponential kernel density estimate.

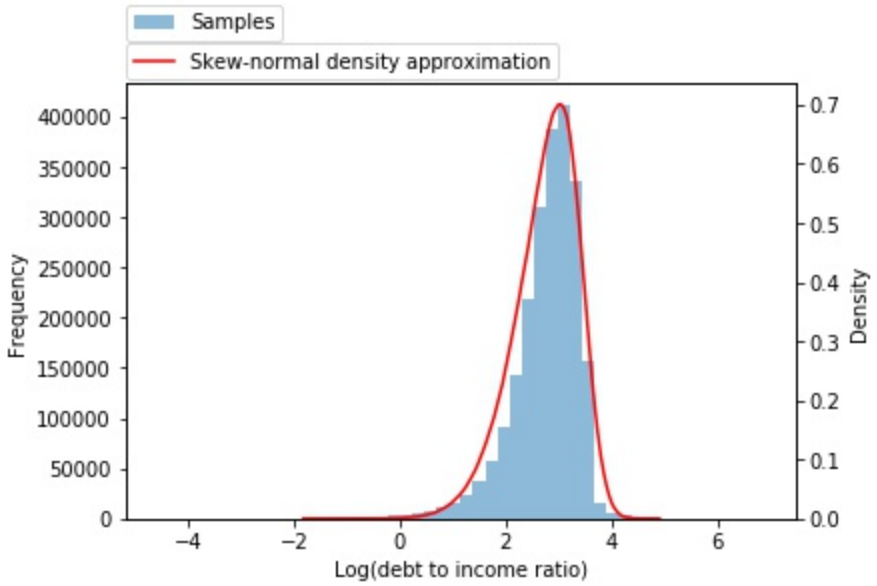

Figure 4. Logarithm of debt to income ratio with skew normal kernel density estimate.

The debt to income ratio is yet another quantity which is close to exponentially distributed has a skewness that says otherwise. Therefore, instead I took the logarithm of the distribution such as to use a skew-normal model for the logarithm.An alternative model for an exponentially distributed random variable Y is to take the log and model it with either a

skew-normal or normal distribution. This is likely a better choice because the skewness of the variates is not what would be expected from an exponential distribution, where it is a constant value independent of the commonly used exponential distribution parameter. In order to apply the logarithm, however, the values equal to zero have to be removed to avoid transformed values of negative infinity; this does not discard too many data samples and so I find it a viable option.

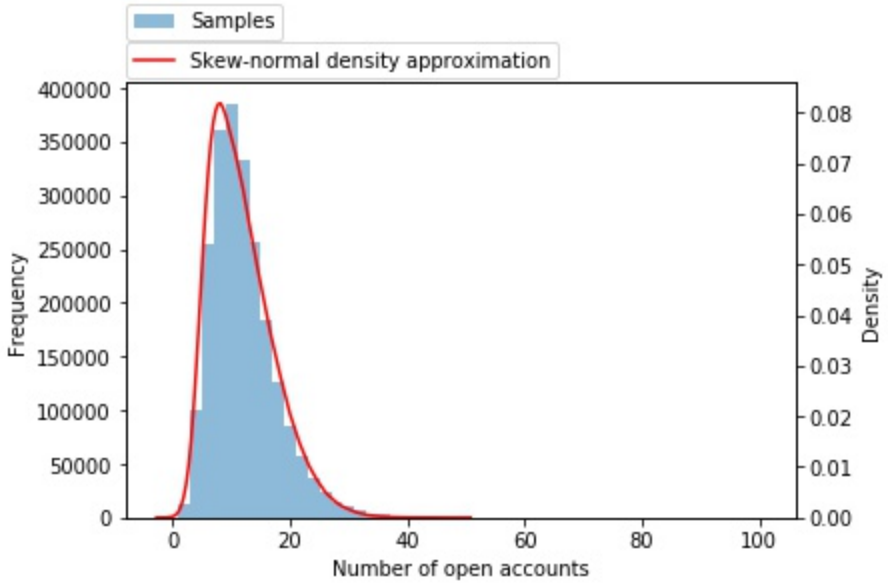

Figure 5. Number of open accounts (discrete) approximated by skew normal kernel density estimate.

The number of open, a discrete valued quantity, nearly looks continuous when plotted as a histogram with bin width greater than or equal to one. This presents an interesting test; given a discrete variable how well can it be modeled with a continuous distribution, taking only the integer part when sampling?

The Kolmogorov-Smirnov test compares cumulative distribution functions of sample and reference distribution (one-sample) or compares the distributions of two-samples. To claim that the approximated kernel density estimates for the distributions are accurate depictions of our sampling distributions, statistical testing is required. The way I set this up is that the null hypothesis is that the distribution is the specified one (with parameters included) and the alternative is that it is not. Therefore, if the KDE plots are accurate, the null hypothesis should be accepted. Unfortunately, this does not happen for any of the supposed distributions, as the p-values were all essentially 0. This means that we reject the null-hypothesis that the distributions were as I had claimed. It seems that the kernel density estimate plots are much more misleading than I thought.

2. In-depth look into correlations between features

		data
fico_range_high	fico_range_low	1.000000
funded_amnt	loan_amnt	0.999999
out_prncp	out_prncp_inv	0.999999
total_pymnt_inv	total_pymnt	0.999996
funded_amnt_inv	funded_amnt	0.999995
open_acc	num_sats	0.999516
num_actv_rev_tl	num_rev_tl_bal_gt_0	0.999125
recoveries	collection_recovery_fee	0.991012
tot_cur_bal	tot_hi_cred_lim	0.972898
total_bal_il	total_il_high_credit_limit	0.951029

Table 2. The top correlations between features in descending order.

There are a number of pairs of features with pearson correlation scores greater than 0.999 for a specific and relatively obvious reason. Specifically, some features are essentially identical; an example being: the funded amount of a loan and funded amount of a loan from investors. If investors represent the overwhelming majority of loan funding then these features are nearly identical which seems to be the case upon inspection. I decided to not include the very highly correlated features as they should contribute little to the modeling process other than computation time. The correlations in their totality can be visualized by plotting the correlation matrix as an color coded image.

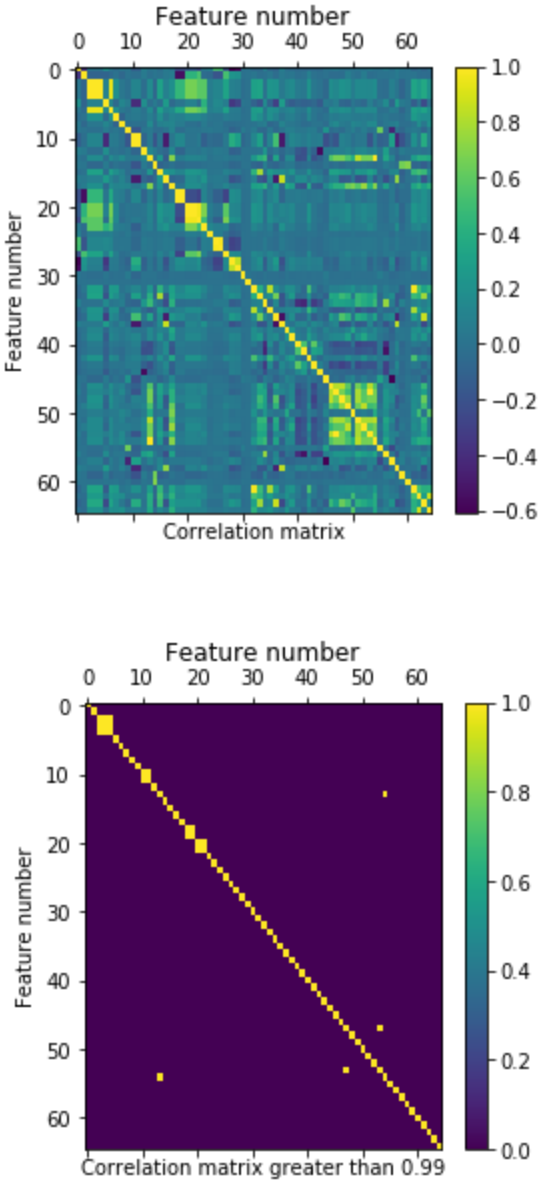

Figure 6. Plotting the color coded correlation matrix, as well as the masked version showing correlations with magnitudes exceeding 0.99.

Likewise, by filtering out all values less than a threshold, chosen here to be a value of 0.99, the highest correlation pairs are demonstrated. Note that the matrix is symmetric so there is redundant information. Because the main targets are the loan status and the recovery amounts, it is prudent to investigate the correlations with these variables. Because the loan status is a categorical variable, it would have to be encoded in order for this to be well defined. Pandas has one-hot encoding via a function "get_dummies" which allows for one-hot encoding. Using this to convert the categorical data to numerical, I could then compute the correlations of this with the

other numerical features. As can be seen, other than the autocorrelations, the remainder of the features have correlation scores between -0.6 and 0.6, approximately. This is sufficient (to me) to not have to drop any of the other numerical features before the modeling process.

Likewise, the same can be computed for the recovered amount of capital. The largest correlation is between the recovered amount and the collection recovery fee; that is, the money that is recovered the more must be paid for its recovery. Therefore, in order to avoid a biased model, I drop the collection recovery fee data from the feature data.With this knowledge I believed that I was ready to begin the modeling process.