Class 2 – EDA on Credit One Data Lessons Learned:

Seaborn can be used as another graphing method apart from matplotlib. Seems to have better looking graphs.

seaborn.factorplot(x=None, y=None, hue=None, data=None, row=None, col=None)

- ***x, y :*** *This parameter take names of variables in data, Inputs for plotting long-form data.*
- ***hue :*** *(optional)This parameter take column name for color encoding*
- ***data*** *: This parameter take DataFrame, Long-form (tidy) dataset for plotting. Each column should correspond to a variable, and each row should correspond to an observation.*
- ***aspect :*** *(optional) This parameter take float value, Aspect ratio of each facet, so that aspect * height gives the width of each facet in inches.*
- ***kind :*** *(optional) This parameter take string value, The kind of plot to draw (corresponds to the name of a categorical plotting function. Options are: "point", "bar", "strip", "swarm", "box", "violin", or "boxen"*

qcut is used to discretize data into equal amounts of bins, so that each bin holds roughly the same amount of data as the next

pd.qcut(x=1D array, q = number of quantiles, equal sized bins, labels=None

Pivot Tables

Pivot tables pivot a spreadsheet, so instead of a column, that column becomes a row or index.

Pd.pivot_table(data = spreadsheet you want to pivot, values = values you want to summarize or work on, index = the rows of your pivot table that your pivoting from column, aggfunct = function you want to perform on values. Default function is average

The problem I've seen with that is that you will then have multi-level columns. To drop the multi-level columns:

Df.columns.droplevel(level=level you want to drop- topmost level is 0)

This well then allow you to work with the columns, for example divide the numbers in one column with the numbers in another column. If you don't use droplevel, you will be unable to do simple arithmetic on columns because you can't do that on multi-level columns.

$$ in Jupyter Notebook Markdowns

Note: in Jupyter notebook, the '$' is interepreted as a MathJax expression. To cancel that out, put two backslashes in front of $: //$

Summary of Analysis Done:

Education vs Default Rates

Graduate School Default Rate: 19.23%

High School Default Rate: 25.17%

University Default Rate: 23.74%

Other Education default Rate: 7.05%

High School Education Rates are the highest. It would seem that the more education, the less of a chance they will default on their loans. 'Other Education' default rates could be excluded from this analysis since it is such a small percentage of the data.

Marriage Status vs Default Rates

Default rates for 1 (Single): 23.46%

Default rates for 2 (Married): 20.95%

Default rates for 3 (Divorced): 26.00%

Default rates are higher for divorced people, and less for married customers, because of the negative financial position that someone can be left in by a divorce.

Limit Balance vs Default Rates

$9999-50,000 31.79%

$50,000-100,000 25.82%

$100,000-180,000 19.87%

$180,000-270,000 16.87%

$270,000-1,000,000 13.77%

Default rates go down as limit balance increases. Default rates are especially high in the first bucket of $9,999-$50,000. This is very significant. There needs to be more scrutiny done on people that apply for loans in this range.

Age vs Default Rates

Age Range	Default Rates
20 – 37	24.21%
27 – 31	19.62%
31 – 37	20.52%
37 – 43	21.70%
43 -79	24.33%

From above, default rate is highest in younger people and in the highest age bracket. It makes sense that the younger people would default more, but why do people aged 43-79 default more? This is something that should be examine further.