Customer requests for an unsecured loan & provides his bank details via Plaid

Accts are matched with Nexis API to remove AML/Fraud cases

No Matches? Check for Red flags in the bank's transaction records {Cash Advance, Excess Activity, Insufficient funds, Pawn shops, overdraft, Credit counselling & bankruptcy, collections}

If there are any hits on the above categories then reject else, move on

If matched with a sketchy account -> Reject loan

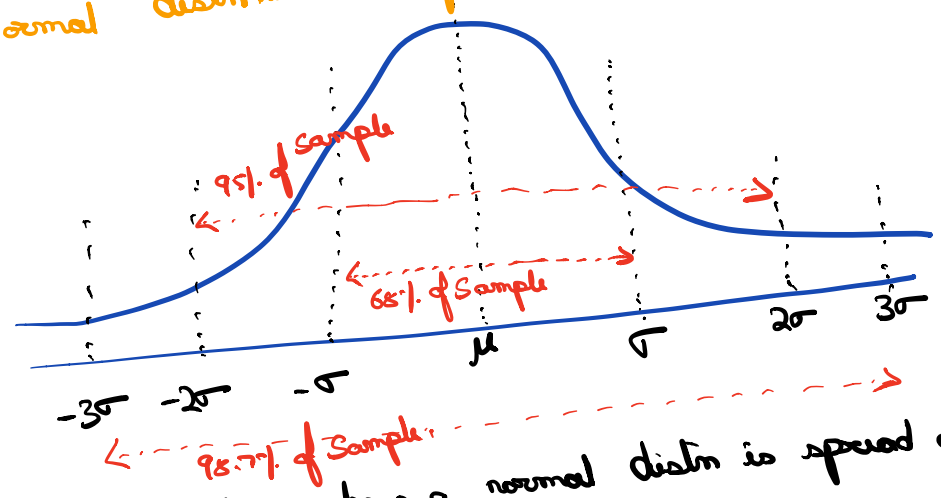
Eqn 1

$$Score = \beta_0 + \beta_1\{\#\ accts\} + \beta_2\{Credit\ card\ utilization\} + \beta_3\{\#\ days\ the\ act\ is\ active\} + \beta_4\{liability/Asset\ Ratio\} + \beta_5\{\%\ of\ risky\ transaction\ in\ Val\} + \beta_6\{Credit\ Score\}$$

where $\beta_0 \ldots \beta_6$ are randomly assigned initially, will explain this at a later point.

Once we get the score, we need to change transform this into a value b/w 0 - 100.

Assumption:- Customers who apply to Teller will have a Normal distn. in terms of eagerness.



Since, we know how a normal distn is spread out, we will be able to guess the σ for the above curve.

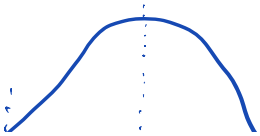
On the first day, when teller product is launched, the 1st customer that applies to our loan will be scored in the following way.

Plotting the customers value in eqn 1 {shown above} we get their score, lets say 456.

Now 456 is the score of the customer

Since the # of sample we have at this point is only 1 the mean also becomes 456

Figure 1



Since we know the distn above, we will be able to determine the σ once we decide the distn curve of the range.

Now the 2nd customer comes to teller & applies for a loan. His score becomes 470. Now the avg of the sample size becomes $\frac{470+456}{2} \Rightarrow \mu$ now becomes 463 from 456

Figure 1 changes -> Instead 456, the new value now becomes 463.

This process keeps happening whenever there is a new application that comes into teller. Every time a new customer comes into the pool. All the existing customers score is re-calculated.

A customer score can change multiple times depending on the number of customers coming into the pool.

In this way a customer score can vary depending on the pool he/she is in.

Transforming the above score b/w 0-100 will be explained later.

Its a relatively simple process

Randomly assigned weights :-

Coming back to the values of $B_0 \ldots B_6$. Initially we assigned them randomly. We did this mainly because we had no data

to model on. Now, once the product is launched & the customers
aboard with our product, we can track them much better. In a one
month's time, we should be able to narrow down on how much
value should be assigned to $B_0 \ldots B_6$. We can do this exercise on a
daily / weekly / monthly exercise to narrow down the actual values of $B_0 \ldots B_6$
& change some of the features altogether, if we consider them not relevant
anymore.