**This Document will have 2 subjects -** R01s decribed with internal data ( 1 - policy) and R01s described with Plaid transaction for January servicing repyments(2 - insights and reccomendation)

\*\*Setting

data\*\* – All **servicing** repayments after 1.10.2023

* filter out repayments showing balance < amount, because of the bug that didn’t check balance for two straight payments but 1 (30 rows with 8 R01s).

[All servicing repayments data query.txt](https://prod-files-secure.s3.us-west-2.amazonaws.com/ee882d05-3de1-4138-a9ca-df906b91724b/78dbfda3-9894-4bd8-a174-f3a4b795daaf/All_servicing_repayments_data_query.txt)

**Solution approach** – Creating features to be used by a decision tree to make policy for preventing R01s.

[R01\_model-All repayments.ipynb](https://prod-files-secure.s3.us-west-2.amazonaws.com/ee882d05-3de1-4138-a9ca-df906b91724b/3c4bcc2a-2152-4775-b222-9cf02dcb0764/R01_model-All_repayments.ipynb)

**Names and Definitions - understand policy rules**

**days since failed payment** – amount of days since the latest user got failed status in repayment.

**balance amount ratio:** balance / amount (rounded 0.05/0.00)**.**

**days since NSF**: amount of days since latest NSF of user in repayment .

**Is on demand**: not scheduled payment

**User R01 prop**: (count R01s) / (count distinct repayments ids) - prior to "this" repayment

**Rent R01 prop**: (count R01s) / (count distinct repayments ids) - prior to "this" repayment

**days from previous repayment:** amount of days since latest user repayment.

**Capital One:** institution name is Capital One.

**days since rent date:** days diffrance with repayment collection and rent transaction rent date (usually the 1st in each loan month)

**appfolio:** Tennent portal is appfolio

**Policy Rules**

**Case 1:**

IF:

User have a **Failed** Payment less ( < ) then 5 days ago

& Is **NOT** Cpital One customer

& balance is less then ( < ) 17,923 (cents)

& **User** NSF proportion is greater ( > ) then 0.5

**Then: don’t attempt to charge R01s amount:  6  R01s propotion:  0.85**

**case 2:**

IF:

User have a **Failed** Payment less ( < ) then 5 days ago

& **Is a Capital One** customer

& More than 28 days since **rent date** ( > 28th)

**then don’t attempt to charge R01s amount:  12  R01s propotion:  0.8**

**case 3:**

IF:

User have a **Failed** Payment less ( < ) then 5 days ago

& **Is a Capital One** customer

& Less than 28 days since rent date ( <= 28th)

& [ [ Is appfolio customer ] **OR** [ age is less then ( <= ) 34 ]  ]

**then don’t attempt to charge**

**R01s amount:  10  R01s proportion:  0.5**

**case 4:**

If days since failed payment > 4.5 days

**&** days\_from\_previous\_repayment <= 2.5

**&** amount\_balance\_ratio <= 1.459

**& Rent** NSF proportion less then 0.1

**&** amount\_balance\_ratio <= 1.061

&  amount\_balance\_ratio > 1.035

**then don’t attempt to charge**

**R01s amount:  8  R01s proportion:  1.0**

**case 5:**

If days since failed payment > 4.5 days

**&** days\_from\_previous\_repayment <= 2.5

**&** amount\_balance\_ratio <= 1.459

**& Rent** NSF proportion less then 0.1

**&** amount\_balance\_ratio > 1.061

& **User** NSF proportion >= 0.15

**then don’t attempt to charge**

**R01s amount:  4**

**R01s proportion:  0.66**

**case 6:**

If days since failed payment > 4.5 days

**&** days\_from\_previous\_repayment <= 2.5

**&** amount\_balance\_ratio <= 1.459

**& Rent** NSF proportion > 0.1

**&** repayment time created and collected are in the same day (within the date on created)

& weekday is 1,2 (Monday and Tuesday )

**then don’t attempt to charge**

**R01s amount:  7  R01s propotion:  0.58**

**Overall policy:**

gets **40** out of 173 R01s with average Accuracy of **~60%+** gets **30** out of 173 R01s with average Accuracy of **~70%+**

**Januray - Plaid Tansactions**

**Plaid Transaction data -** restriction - all transaction only have the date transacted (not hour)

[January R01 users TRANSACTIONS.txt](https://prod-files-secure.s3.us-west-2.amazonaws.com/ee882d05-3de1-4138-a9ca-df906b91724b/47f548d9-7134-4dfd-8ecb-75f00bfa2582/January_R01_users_TRANSACTIONS.txt)

[Jan Transaction manual check.txt](https://prod-files-secure.s3.us-west-2.amazonaws.com/ee882d05-3de1-4138-a9ca-df906b91724b/0bb1d2b1-d1f7-4f59-9e76-aed5e416da93/Jan_Transaction_manual_check.txt)

* most R01 transaction showed either other **Loan payment** or **between accounts transactions** in the day of collection.

\*\*\*\*out\_transactions = all the money spent by user ( in collection day ) in\_transaction = all the money earned by the user ( in collection day )

using plaid i first tested if in retrospect we can see that the user wont be able to pay the payment based on plaid transactions, the amount . and the plaid balance at the time.

* **33% out of all R01s has** → [balance] - [out\_transaction] + [in\_transaction] **<<** [amount] meaninng that **67% shouldnt not** get R01 at the end of collection day.

Not having accurate time transacted made me take a worst case senerio view - lets say all the money earend by the user only was at the very last moment of the day ( after repayment) so that now we are over looking all in\_transactions

* 7**5% out of the R01s has** → [balance] - [out\_transaction] << [amount] meaning we can **understand those R01** if the earning were not realy put into the account prior to the repayment…

\*\*Optional data for further investigation

with plaid transaction data on all January repayments users (not just R01) we could create and test features based on Plaid Transactions data such as:\*\*

ratio money in / money out , ratio monthly spent / monthly rent amount , amount and count of in between accounts transactions ( how much money user "bounces" between different accounts), days since income , other loan payment ratio – other loan payments amount / our amount , amount of other R01 gotten , ETC…