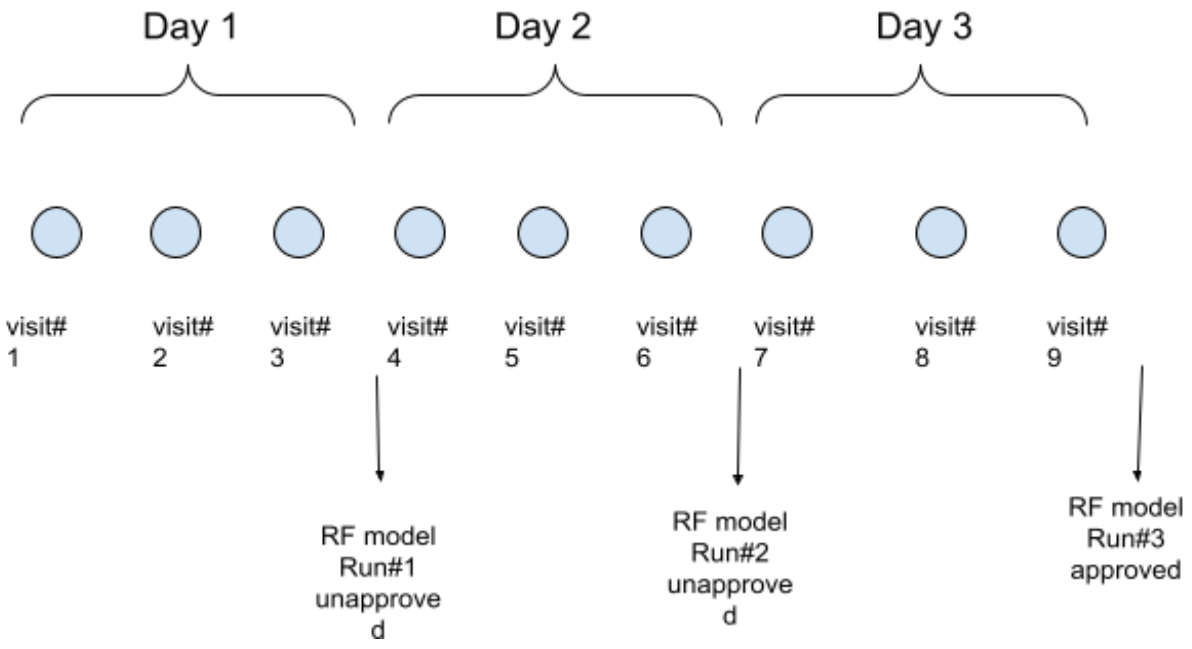
Real Time Approval

Authors: vivek.pandey, nilesh.mp, praveen.yadav
March 2022

What is Real Time Approval?

Currently, “approval” or underwriting is done by an airflow job which runs on a nightly basis. This job runs “challenger model” (which is a random forest based model) on all the users who were tried to transact the previous day, but were found to be not approved. Some of such users are approved by the model and this is noted in a table.

So, a user journey might look like the following:



Currently RF model runs nightly and approves appropriate users. In the above diagram, a user is not approved in the first two runs of RF model, but is approved in the third run.

Note that since user is unapproved at the end of visit#6, but is approved at the end of visit#9, it might be the case that user could have been approved had the model run after visit#7, or visit#8. Thus, with nightly runs, we lose the transactions corresponding to visit#7 and visit#8.

This loss is what real time approval aims to eliminate.

Potential jump in approval rate

We did an exercise to estimate the potential increase in approval rate on implementing RTA.

Following data is on a unique user basis (i.e. not on the basis of approval call basis).

Category (A)	% users falling the category (B)	Current approval rate for the category* (C)	Approval rate for the category with RTA* (D)
Last seen in 0-15 days	62	61	73
Last seen in 15-30 days	10	50	62
Last seen in 30-90 days	11	43	58
Last seen in 90-180 days	3	33	52
Last in in 180 days - 1 year	3	20	69
Last seen more than 1 year ago	6	3	53
Never seen (i.e. new user)	4	0.09	92

* These numbers are at a threshold of 0.3.
Current approval rate = average of column (C) weighted by column (B) = 50%

So, expected approval rate after RTA = average of column (D) weighted by column (A) = 68%.

So, we expect approval rates to go up from 50% to 68% (on a unique user basis)

At the threshold of 0.25, RTA approval rate drops from 68% to 61%, still providing us substantial gains.

Architecture

This diagram provides high level architecture. There are two main components:

1. Realtime event pipeline: It reads events from kafka topics, aggregates them, creates model features and stores them in dynamodb

2. RTA service: It is the service called by transaction system to get approval status of a user. It reads features from dynamodb (potentially collates them with the features got in the request), runs ML model on these features, and returns the result to transaction system.

Tasks and Timelines

Overall: v1 by Apr 27, v2 by May 6.

Area	**Component**	**Comment**	**Time required**	**Finish date (Tentative)**
Pipeline	Complete the functionality of UserActivityData feature. (Some new features have come up lately, we will exclude these features in v1; if some of them are easy, we can do them)	Need to discuss with DS team	5 days	March 25 [DONE]
	Complete CI/CD build pipeline to deploy in staging/production	Permission issue with code factory EMR accepts jar instead of docker image	3 days	March 31 [DONE]
	Flink deployment topology	no of workers memory footprint elasticity (increasing/decreasing workers)	2 days	Apr 4 [DONE]
	Complete and test end to end functionality on staging. Check that all events are processed and data is aggregated.		5 days	Apr 11 [DONE]
	Monitoring pipeline for delays and dependencies	Ways to identify if pipeline has stopped or erred	5 days	Apr 18

RTA service and dependencies	Deploy and productionize CarrierData service	Creating CI/CD pipelines Identify H/W requirements	2 days	Apr 20
	Containerize model (Challenger v3) as an HTTP service	Model take care of the case when it is first call and there is no data in dynamo DS team to work on this	4 days	Apr 15 (parallel effort)
	Develop RTA service: Basic	Use only dynamodb data	5 days	Apr 27 [DONE]
	Develop RTA service: Advanced	Merge call data with dynamo db data	5 days	Do this in fast-follow v2. Finish by May 6.
Transaction system call	Transaction system should call RTA service	To be discussed with transaction system		Apr 15 (parallel effort)

Outcome of discussion on March 14:

1. Can some of the following be left out in version 1:
    a. city as a feature [city is important, we receive latlong from merchants for only 10% users]
    b. carrier as a feature [Let's include carrier information]
    c. merging call data with dynamodb data [v2]
2. Approval model will also compute credit limit. Does that introduce any new complications? [TO BE DISCUSSED]

How will credit assignment/credit line work? Need to be figured out.

Pending tasks [May 24, 2022]

1	We sometimes get multiple approval events. We need a mechanism so that we don't process the same data twice.	Abhinav	Nilesh to sort out details. Completed the design and have updated the diagram (tab: RTA-Flink)

	DE-486		PR review done. Waiting for Gurpreet to set up redis on prod. (By mistake, Nilesh asked it for staging). Gurpreet needs sizing details. Nilesh to provide sizing details.
.2	During deployment, cancellation of existing job has to be handled DE-460		Not a blocker for release. Without this, there is manual toil each time there is a deployment.
3	Monitoring for pipeline		
4	Deploy and productionalize carrier data service (For RTA service) (DE-494)		Nilesh: We have dropped this feature from our RTA feature list. [May 30] We need this [May 31] Need to discuss with Praveen. Have a call at 4:30 PM. [June 2] carrier data service is a part of RTA service itself. PR has been raised and Abhinav has approved it.
5	Deploying RTA as a service	Abhinav	In progress. Maybe 1-2 days. ETA: May 30 [May 31] This is done
6	Model as a service		Vinayak + DS to work on this. Praveen knows that there is DS work on this. If Vinayak is done then we can unblock Praveen. [Vinayak to give ticket number] [Jun 2] We have deployed in staging. [p50 is 24ms] [June 7] p99 is 37ms with concurrency = 3.
7	Improve MaaS performance		
8	Deploy MaaS in prod		[June 14] Nilesh will check with Vinayak.

9	Integration with TS		Open question: We only have production RTA as a service. What will staging TS call? Nilesh to find what to do here. Nilesh: Had a call with the Txn team and have agreed on RTAService being in production while the consumer is on staging. [June 2] Unanswered question around: should dRTA replace sRTA?
10	MaaS will return PD. RTA as a service needs to find approval line + gcl based on this PD.		Nilesh to find how to find the gcl + approval line. *Nilesh: After discussion with Praveen, it is agreed that the model is handling this.* *[DONE]*
11	Backfill - One of the challenges is the exact cut-off date-time.		Nilesh to come back on exact logic for cut off data-time after discussion with Praveen. I had a discussion with Praveen and we are good to go with an overlap of a day at max. The implementation approach details are in this jira ticket Abhinav starts ~June 3 [June 2] Gurpreet to help with Kafka topic problem.
12	RTA service performance testing		We will do this staging dynamo is ready.

[May 31] Gurpreet said that he will talk to Shoan about new kafka topic for backfill. Nilesh & Gurpreet are meeting June 1 morning to discuss the same.
[May 31] About staging ts connecting to staging rta service. DE can deploy staging rta service. Gupreet has said that he may be able to replicate prod dynamo to staging dynamo. This is also be discussed on June 1.
[May 31] Vinayak & Nilesh said that Praveen & Harkirat should be done today.

[June 2] Nilesh to unblock Gurpreet by providing sizing details.
[Jun 2] Vivek to get in touch with Sam and find the prioritization

Challenges for backfill
==========
[June 7] We will have a separate kafka cluster (we don't want to put a high temp load in prod kafka). Praveen Raghav will provide this to us. Should return by June 7 EOD.
[June 7] There will be a separate flink cluster. Nilesh talking to Gurpreet. Might get this done by June 7 EOD.
[June 7] For backfill, we will (tentatively) go ahead with a session length of 1hr across merchants. Need some change in flink code. Need retraining of the model. (DS may already have trained it)

Tasks for Devops [Nilesh to follow up]
==========
1. Kafka cluster for backfill
2. Flink cluster for backfill
3. Staging dynamo

[June 14]
Dev ops tasks
==========
1. Set up backfill flink cluster. Gurpreet is working on it.
2. Set up staging dynamo. Gurpreet will work after (1)
3. EMR notebook cannot connect to s3