ML Town Hall

Introduction

* Hi all, I’m Simon and I am going to take you through one of our project which is the intelligent payment routing project.
* I’ll first give some background of the project, problem statement, potential benefits of the project and proposed solution

Background

* For international swift payments, which goes through GPP UK payment system, it would need to go through a number of intermediary banks.
* In the middle, shows an example of the stages of the transaction as a client sends a payment to a company in America
* The routing is decided by a number of rules set in GPP, but the first one is always to see if Barclays has a direct relationship with the intermediary bank
* We have access to the transaction in GPP UK but we will only see the start and end of the transaction and not the individual parts

Swift API

* To get the full end to end enriched routing data, we will need to use the Swift API GPI Tracker
* On the left is an example of the response you get when you hit the api
* You can see it will have details at each of the payment events i.e. the intermediary banks
* To the right, show the details when we expand one of the payment event
* We can get details such as what parties are involved
* What were the charge amount
* And how long it took to process the payment.
* So you can see with this enriched data we can gain a lot of insight into swift transactions such as which intermediary bank takes the longest to process transaction, or which one charges the most

Problem Statement

* As you saw, the enriched routing data can be very useful and powerful
* An issue we found from starting this project is that this enriched data is not saved down anywhere in the bank.
* Only a select few client facing staff have access to the API and they can only search one transaction at a time.
* Clients can track their swift payments through the track and trace service through iPortal, again they can only check one transaction at a time.
* Therefore, with no easy access to the routing data, we often find that we cannot easily answer clients queries around their transactions.
* For example why is their USD transactions taking longer than expected.
* Therefore, this projected isn’t only about optimising the routing but also giving users the enriched data to easily answer clients queries.

Hypothesis

* On this screen is our hypothesis tree, what we believe will be the benefits of the project.
* Starting from the left, we believe by enriching the transactions data with the routing data it will improve data visibility
* The top path is the optimisation problem to find the optimal route for fees and speed.
* The bottom path is for the client MI which will improve client servicing
* Some of the benefits from this project is improved efficiency, improved client retention, increased revenue from competitive routing options

Approach

* With most of our projects, our project lifecycle usually start with a proof of principle which has quite a limited scope and is supposed to be short and sweet
* If that is successful, then we will move onto a proof of concept which will expand the scope further
* Finally, if required we will then productionalise the model.
* For the project, our proof of principle the main part is checking if it is technically feasible to get the data from the API
* Our plan is to get the routing data for a particular client, looking to USD payments to America for one month of data.
* When we do get that data, we can analyse the data for optimal routes in terms of fees and speed.
* We will also create a tableau dashboard to surface the insights to stakeholders.
* If the proof of principle was successful we will then expand out the scope to all clients possibly looking at 12months of transactions
* I am currently at the the PoP stage and unfortunalty we had to put the project on hold as it was very difficult to obtain the routing data from the API as there is no easy efficient way to connect to it from Barclays.

Proposed Solution

* Although we do not currently have the data yet
* I believe we can treat this as a network problem, where we treat each bank as a node
* And the edges are the payment flows, where the edge attributes are the fees and processing times.
* We can then apply shortest path algorithm to find optimal routes
* At first instance, we will try a greedy algorithm and see how well it does and also try some other techniques such as reinforcement learning