OneMain Financial

Northwestern MSiA and Kellogg

October 2020 - June 2021

Agenda

1. Executive Summary
2. Project Structure
3. Understand User Flows
4. Identify Frictions
5. Predict Likely Outcomes

Project Structure

Team

Northwestern MSiA Team

Zhiyang (Iris) Chen

Hao Xu

Chengcheng (Chelsea) Zhang

Congda (David) Xu

Northwestern MBA Team

Kayla Grayson

David Crosson

Kevin Bay

Project Goal

Identify “frictions” in **current customer experience** and predict application outcomes, with the goal of optimizing workflows and **increasing user engagement.**

Project Goal

Identify “frictions” in **current customer experience** and predict application outcomes, with the goal of optimizing workflows and **increasing user engagement.**

1. **Understanding user workflow:** understand and decompose the action flow that a customer needs to go through to make a submission.
2. **Identifying frictions:** define “frictions” and identify primary KPIs based on impact on application outcomes; prioritize metrics that will ‘move the needle’ as focus of ongoing monitoring
3. **Predicting likely outcomes:** build machine learning model(s) to predict likely outcomes of a potential borrower as they move through the OMF application

Project structure covered in three distinct phases

Phase 1
Understand user flows

Understand OMF's status quo and business situation

Prepare data
- Data cleaning
- Further data exploration
- Understanding sources
- Feature engineering

Understand user flows via:
- Heap
- Manual Mapping

Phase 2
Identify frictions, KPIs

Hypothesize drivers of friction

Identify frictions
- Qualitative analysis of user recordings
- Quantitative analysis through Heap funnel

Obtain metrics of interest, visualize data in dashboards

Phase 3
Build model

Select model
- Test different models
- Compare model outputs
- Select best-performing model

Train final model
- Train model with complete dataset
- Validate model results

Create intervention methodology

Three core data sources power the analysis

Focus of this project

OneMain Financial	HEAP	hotjar
Landable events	**User events**	**User recordings**
Original data source	Primary data source	Basis for qualitative analysis based on recordings

Landable event data for individual-level real-time analysis

OneMain Financial

Landable events

Main data source

- Records system events that are triggered by applicants during applying process
- Linked to application details, device information and demographical data
- Potential drawback: landable events are not directly linked to application activities on the pages

Heap Data for aggregate-level analysis and dashboard

HEAP

User events

Primary data source

- Has data on detailed user behaviors such as clicking, changing entries, time spent, etc.
- Can compare with historical data (previous week/ month/ quarter/ year)
- Can group by qualitative characteristics to learn about different user segment
- Drawback: can only access aggregate-level data through Heap platform

Hotjar analysis to complement quantitative analysis based on landable events

hotjar

User recordings

Basis for qualitative analysis based on recordings

- Review recordings of users and recognize patterns qualitatively
- Cluster observed patterns and make UX recommendations based on insights
- Form friction assumptions for Heap and landable event data validation

HEAP data provides better insights into user behavior than “Landable events”

OneMain Financial

DATA

User opens form

User opens terms window

User clicks on ZIP lookup button

User submits form

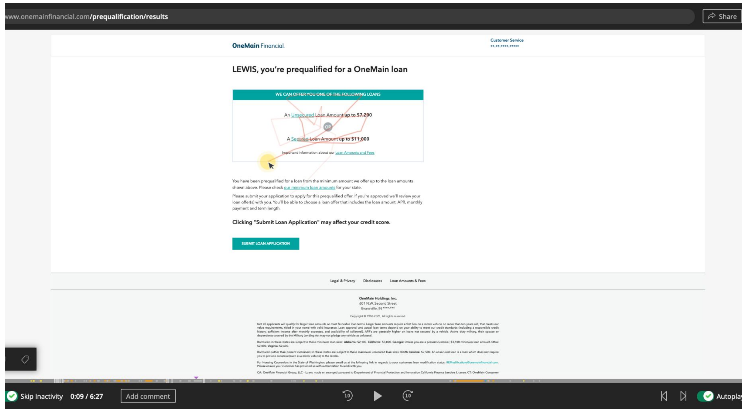
HEAP

DATA

User opens form
User clicks into first field
User takes 2 seconds to type name
User clicks into last name field
User takes 3 seconds to type last name
User clicks into ZIP field
User cannot find ZIP
User retypes ZIP
User needs 74 seconds before moving on
User types desired amount
User changed desired amount
User took 4 seconds to read terms
User confirms terms and conditions

hotjar

VIDEO RECORDINGS



Phase 1

Understand User Flows

To understand user flows, Markov Chain Approach was utilized

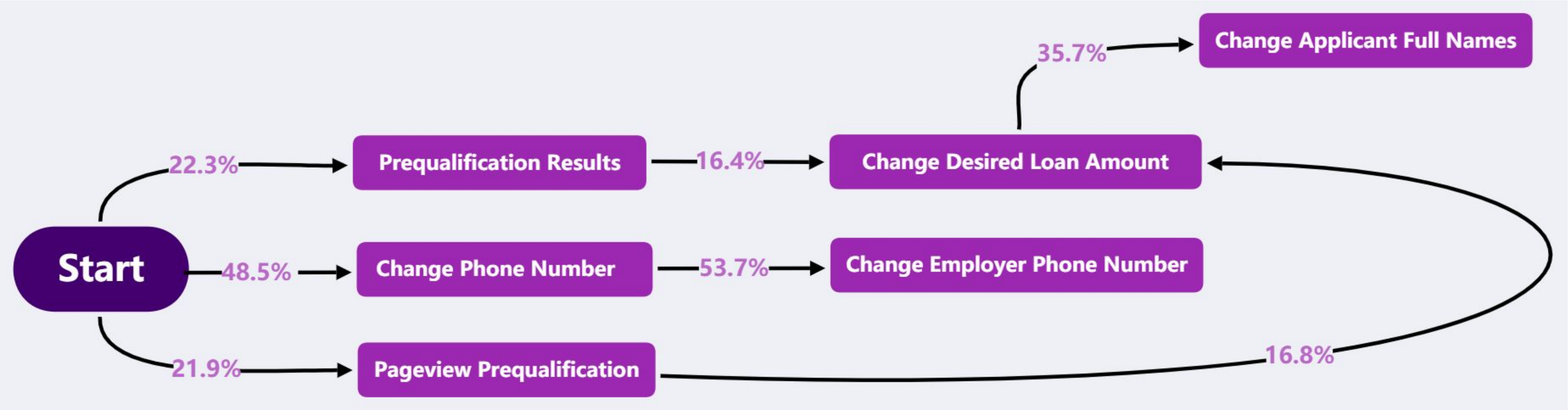
What is a Markov Chain of user flow?

- An automatic mapping of user flow from user action data
- Find transition possibility among major steps - automatic funnel generation
- Calculate average time spent on each possible transition

What can Markov Chain do?

- Provide guideline for friction definition and model building
  - Outputs from Markov chain can be used as reference for KPI defining
  - Thorough understanding of user flow is required to build action chain based model
- Provide benchmark to trigger potential online customer assistance
  - Expected time to spend calculated for each step can help identify users that need help

Markov Chain Demo



Key Findings Match Intuition

Heap

Top stages that have the longest expected time:

- click_look_up_my_offer: 118.86s
- click_loan_amounts_and_fees: 75.09s
- click_loan_purpose_medical_expenses: 54.82s
- click_dm_link_mobile: 48.40s
- submit_check_for_prequalified_offer: 45.35s

The most possible ending points:

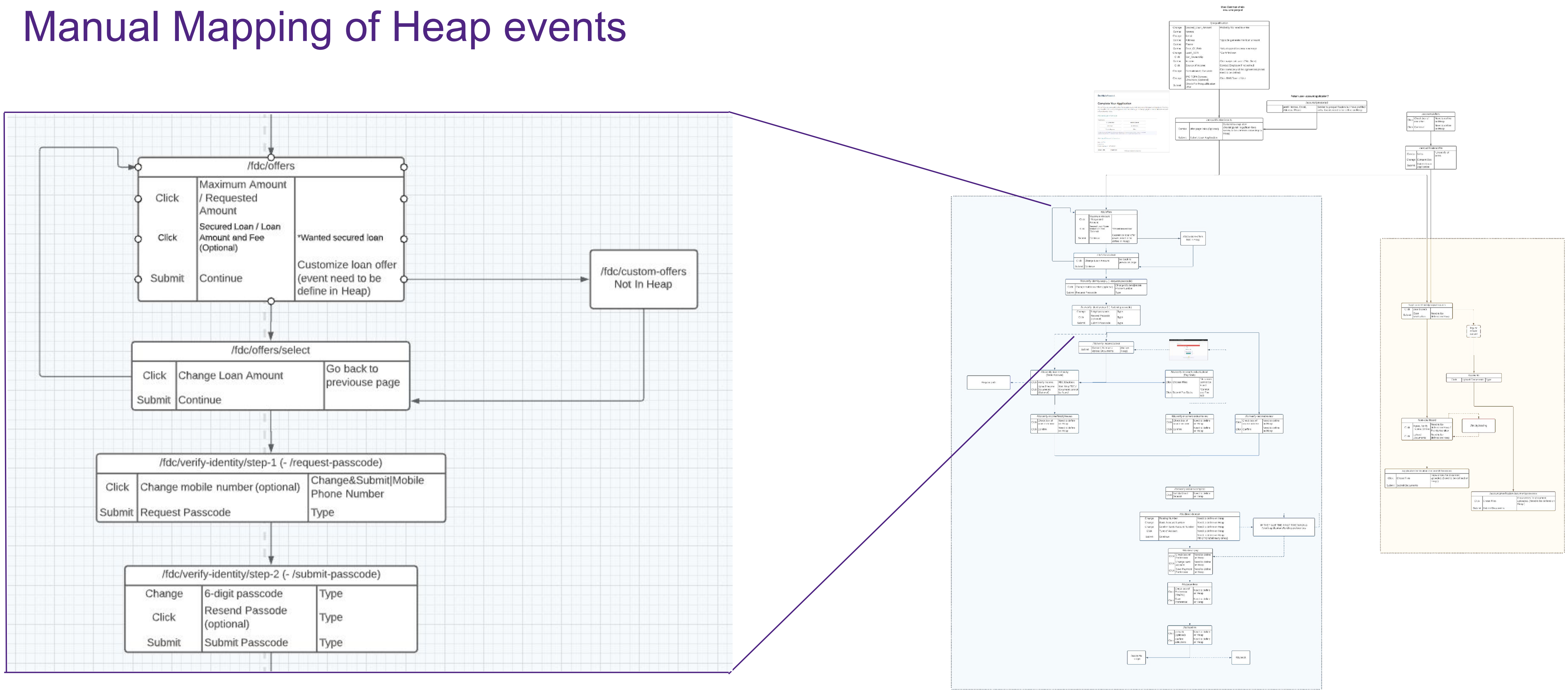
- click_loan_amounts_and_fees
- click_checking_saving_both

Intuition

- The stages that have the longest expected time are generally those that require users to spend some time reading the information, such as offer and fees.

- The stages that are most likely to be the ending point of a action chain also include reading and checking information, which takes time.

Manual Mapping of Heap events



https://lucid.app/lucidchart/ba349a6c-60e2-48cf-acdb-43a377954c2f/edit?shared=true&page=0_0#

Phase 2

Identify Frictions

What Creates Friction: Ingoing Hypotheses

Information not on hand

Some inputs may require user to look up details they do not have readily available:

- Salary
- Social Security Number
- Current Employer Phone #

Discouragement/ Discomfort

Users may reconsider moving forward when asked to share sensitive information:

- Salary
- Social Security Number

Information provided from OMF may discourage users from continuing:

- Loan Fees
- Min/Max Loan Amounts by Zip Code

Errors

Error messages may increase the likelihood of user friction:

Potential User Errors:
- Incorrect inputs due to unclear instructions
- Error triggered by typo

Potential System Errors:
Crash
Slow Response

Friction can be characterized based on multiple inputs

1. Drop rates

User drops at certain pages or certain steps can indicate serious friction points that cause users to end their current process

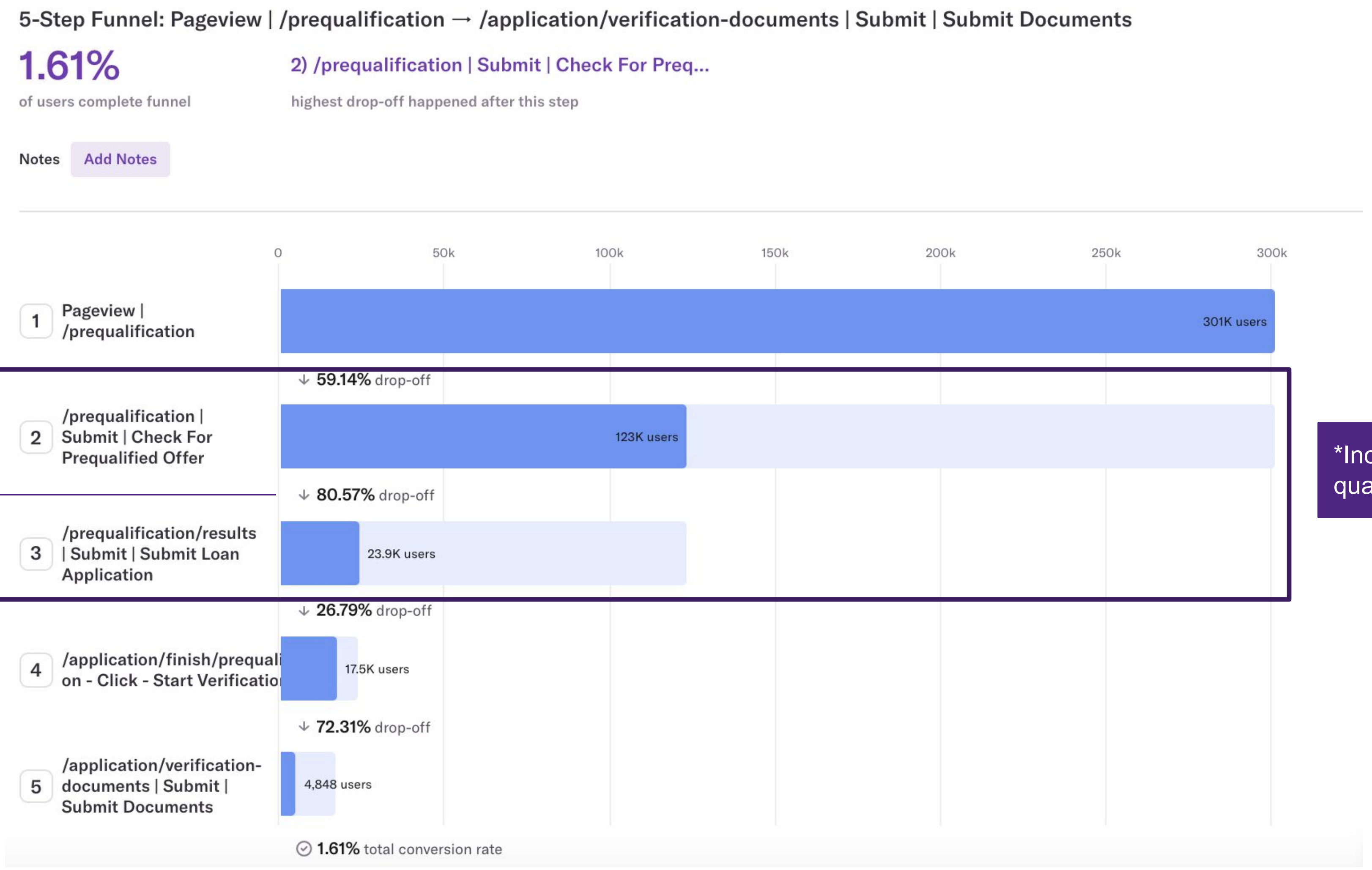
2. Time spent

If a user spends a significantly longer time on a certain action or overall needs more time to complete

3. Events triggered

Certain events, while still happening on the page, can indicate that a user has experienced a point of friction (such as using the back button)

~81% users dropoff between **checking pre-qualification** and **submitting loan application**

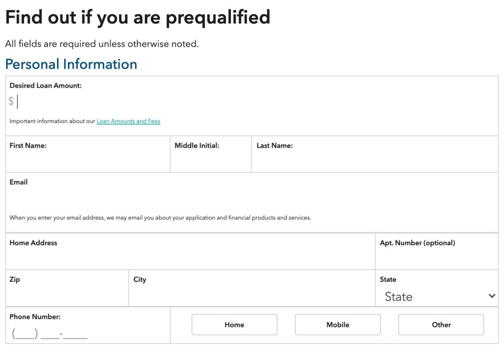


80.57% drop

*Includes users who are not qualified for an offer

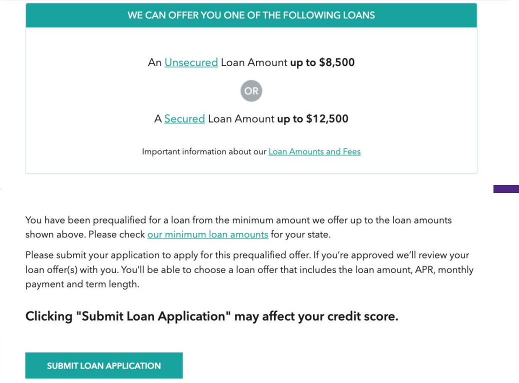
https://lucid.app/lucidchart/ba349a6c-60e2-48cf-acdb-43a377954c2f/edit?shared=true&page=0_0#

On average, users take ~33 minutes to finish the full application flow



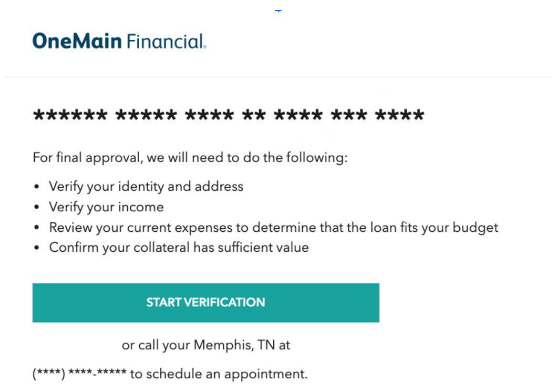
/prequalification

3m5s



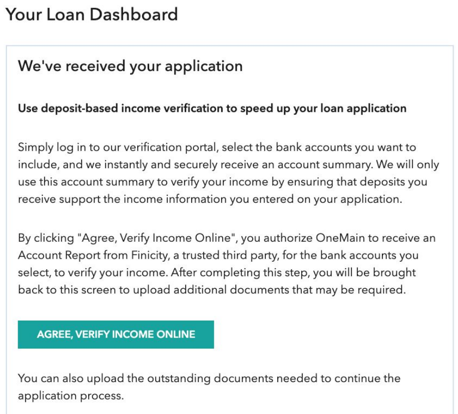
/prequalification/results

2m18s



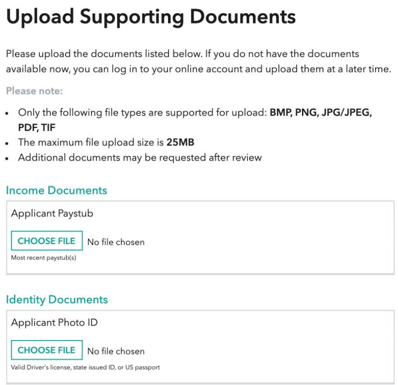
/application/finish/prequalification

18m55s



/loan-dashboard

4m42s

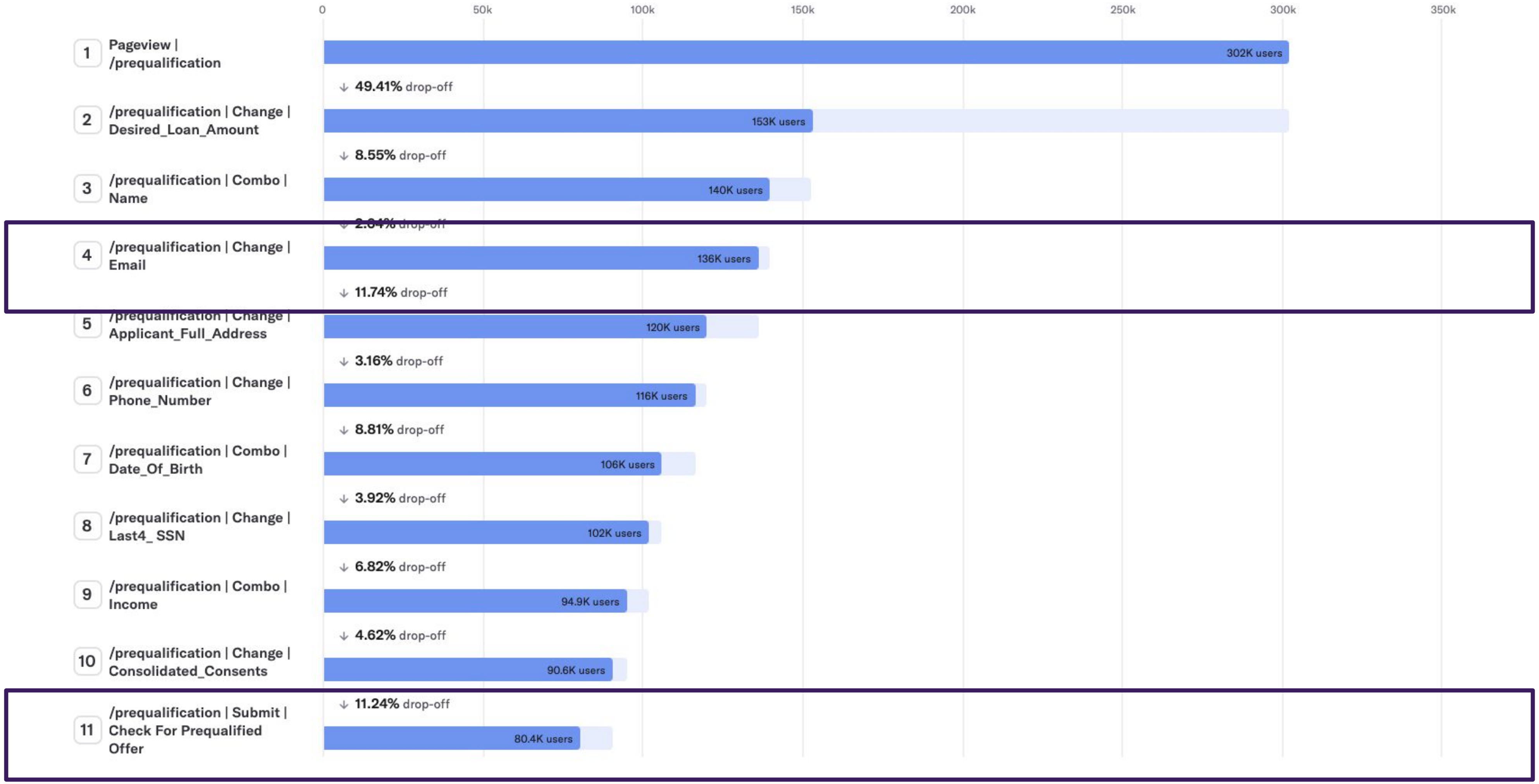


/application/verification-documents

4m9s

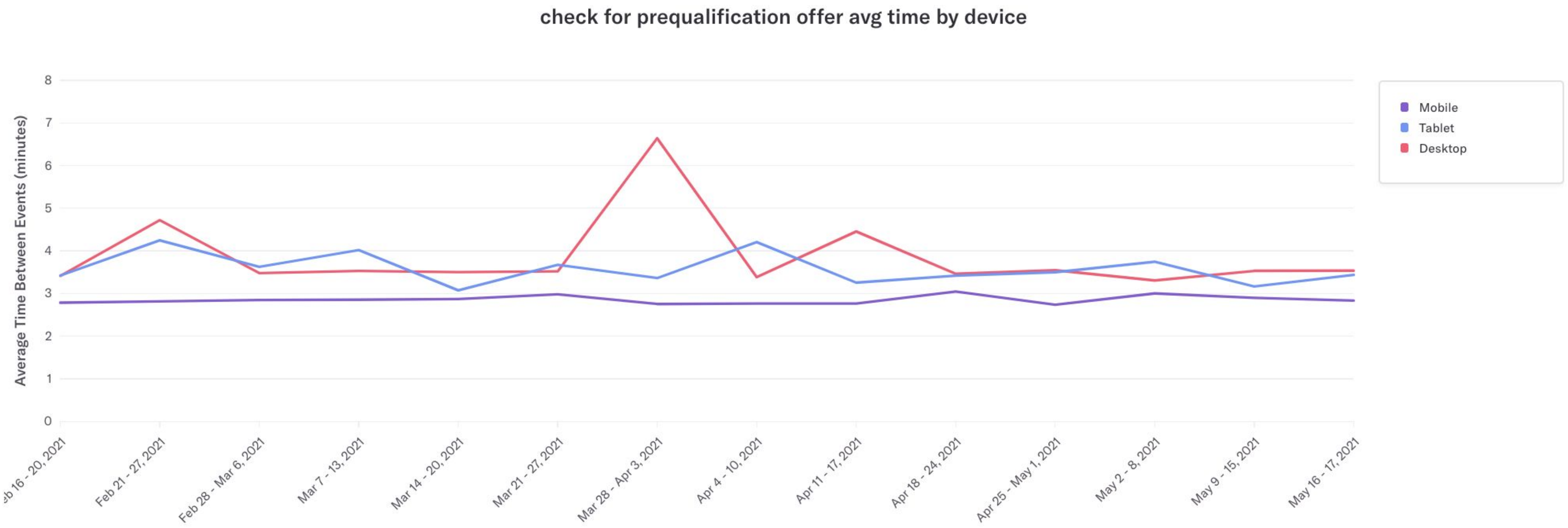
*Data: past 90 days

Zooming in, we see substantial dropout before entering full address and again after checking the consent box but before submitting



https://heapanalytics.com/app/env/2104307948/funnel/report/prequal-fields-1376877

Mobile users take the least amount of time and are the least volatile



https://heapanalytics.com/app/env/2104307948/graph/report/Average-Time-Between-Pageview-prequalification-and-prequalification-Submit-Check-For-Prequalified-Offer-grouped-by-1-property-1376827

Following qualitative analysis of Hotjar recordings, Heap served as a rich source of insights on consumer behavior

Noteworthy points of friction in the customer experience:

- Reluctance to share address
- Large portion of users click ‘I agree’ but then drop off
- Many users are going through the prequalification process multiple times (on average 2)

As a result, we recommend regularly tracking the following metrics:

1) % Users dropping between Email and Full Address
2) % Users dropping after check for prequalification offers (currently, 59.14% drop off)
3) % Users discontinue after viewing the offer results (currently, 80.57% drop off)
4) Average # of attempts (calculated as ratio of submit button clicks/unique users)

Phase 3

Predict Likely Outcomes

Data Sources

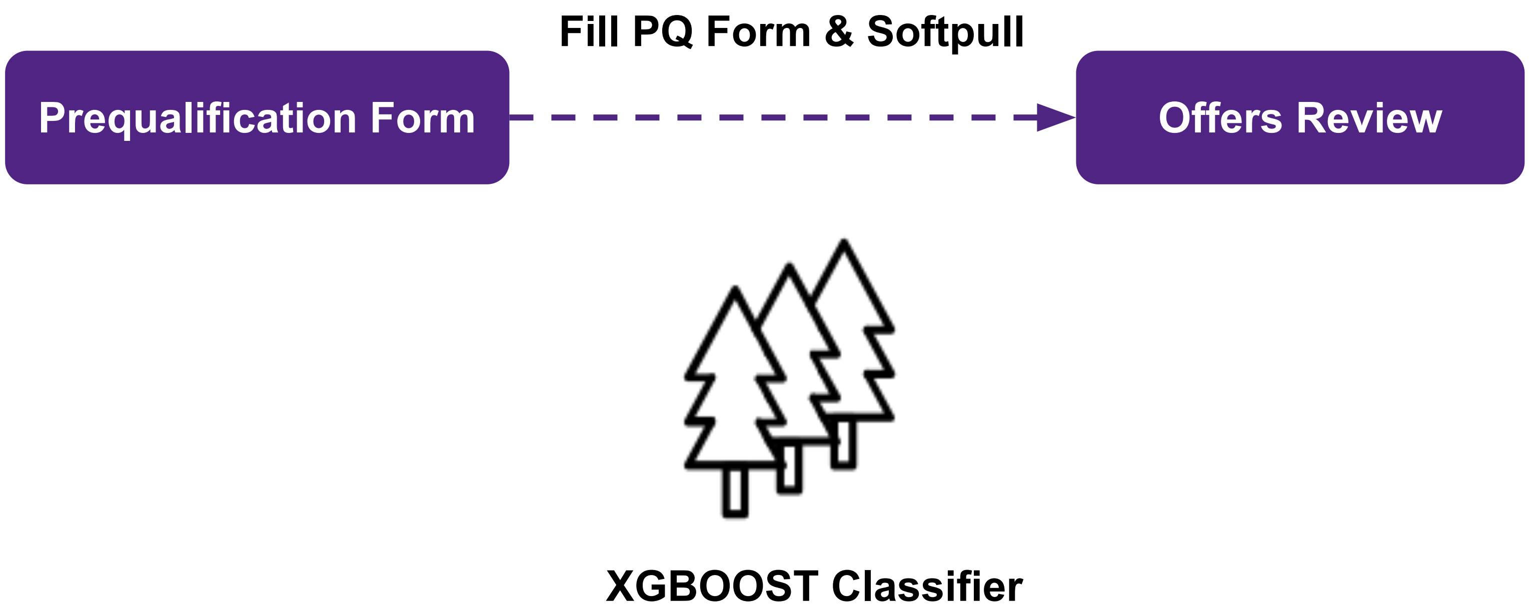
OneMain Financial

Landable events

HEAP

User events

Boosted Tree approach handles non-linear processes accurately and efficiently



How it works: User behavior generates probability of advancing to next step

Input Data (Represents one user)

Snapshot	Accum. Time	Event 1 Count	Event 1 Time	Event 2 Count	Event 2 Time	...
0	0 s	0	0 s	0	0 s	...
1	5 s	1	5 s	0	0 s	...
2	6 s	2	6 s	0	0 s	...
3	26 s	2	6 s	1	20 s	...
...	...	...	...	...	...	...
n	477 s	15	212 s	12	27 s	...

XGBOOST Classifier

Prediction

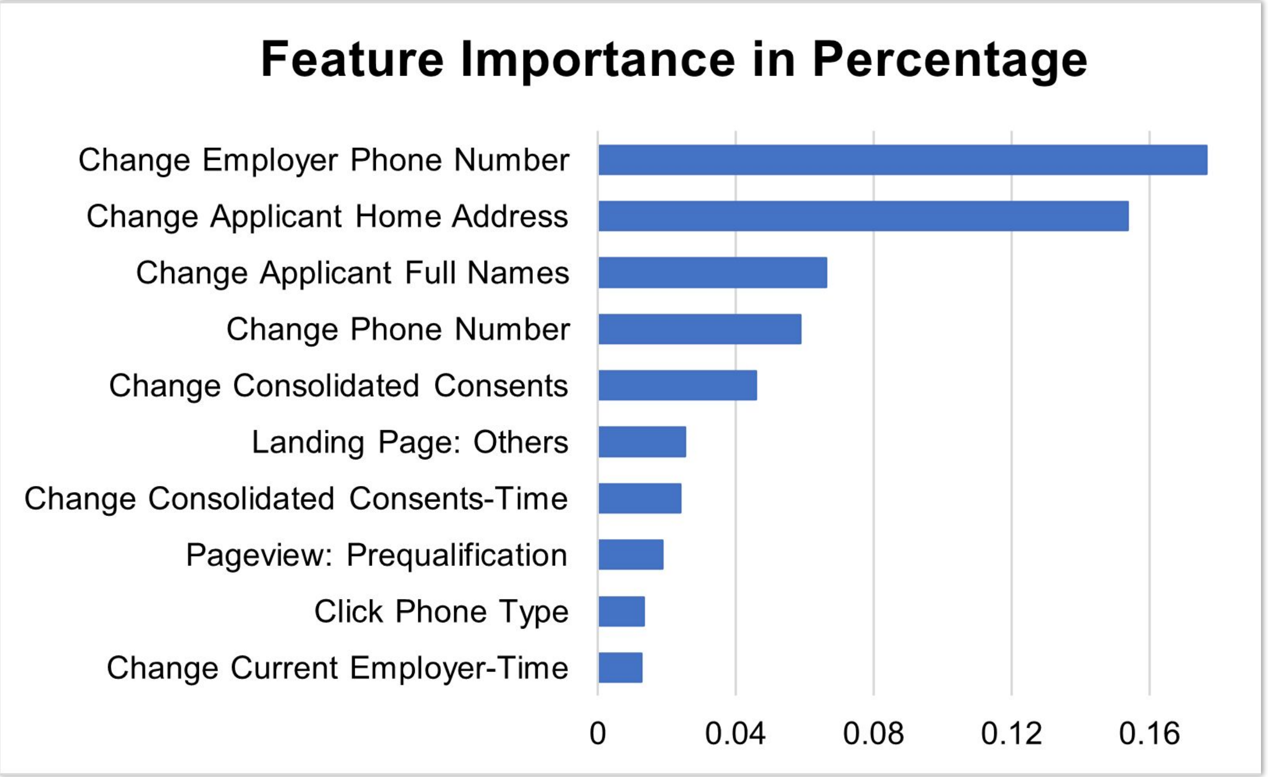
	Prob. to Next Stage
0	0.37
1	0.46
2	0.63
3	0.34
...	...
n	0.13

Model predicts drop behavior with 83% accuracy; time and events triggered emerge as potential points of intervention

Accuracy:

Overall Accuracy: 83.58%

No Information Rate: 71.96%



*No-information rate: The proportion of the majority observed classes. This is used to compare with the accuracy rate to see by how much our model beats a naive guess.

Demo

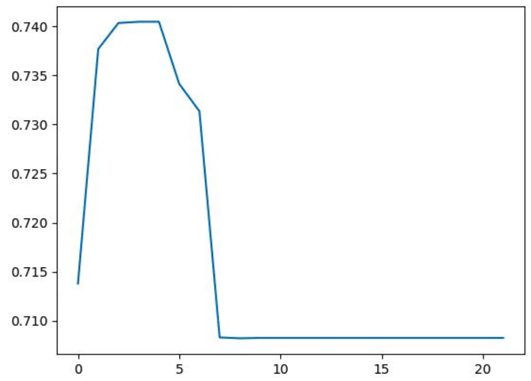
Value of Predictive Model

1. Closer look on how each event affects the likelihood of finishing the application

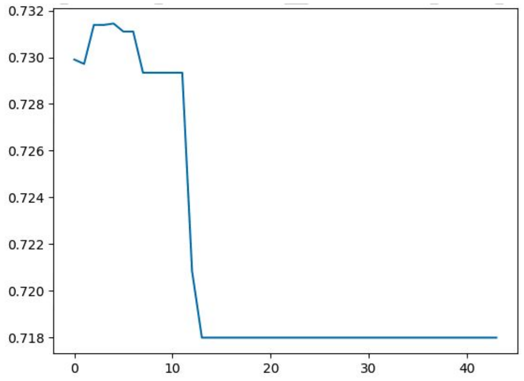
2. Optimizing the intervening mechanism

Deep dive into features shows differentiated impact on likelihood to submit

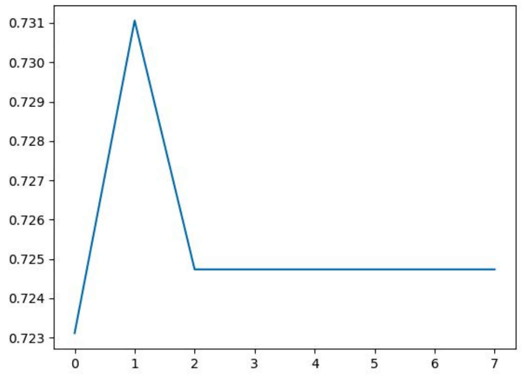
Change Full Name



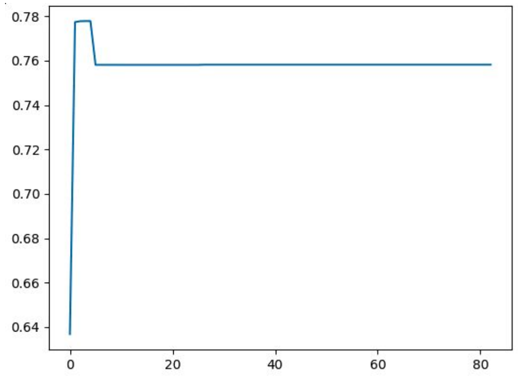
Change Contact



Change Zip

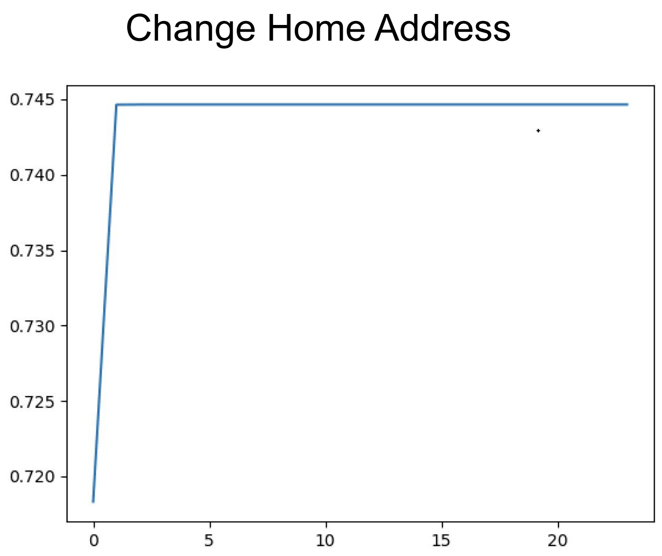


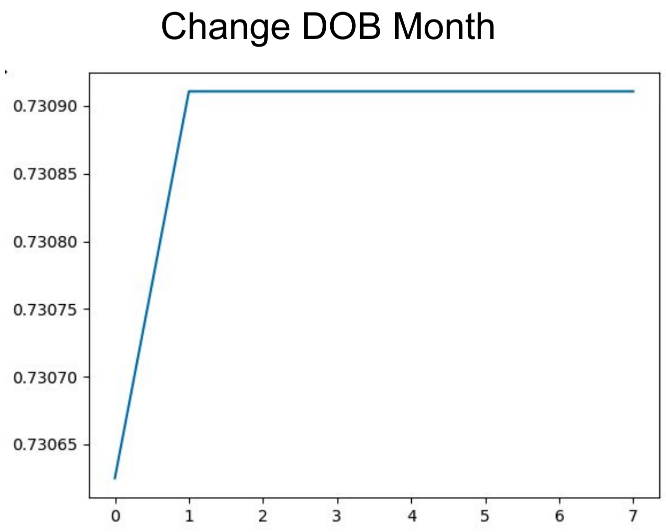
Change Employer Phone



- There is a general pattern for all the important features - since it’s sequence data, we are expecting to see peaks like these
- Peak point can be used as references for KPI benchmarking

Deep dive into features shows differentiated impact on likelihood to submit





- This pattern indicates the event itself doesn't have significant impact on likelihood but whether applicants can reach to it matters a lot
- The previous steps here are providing email address and providing phone number respectively, this is inline with the conclusion from analyzing Heap data

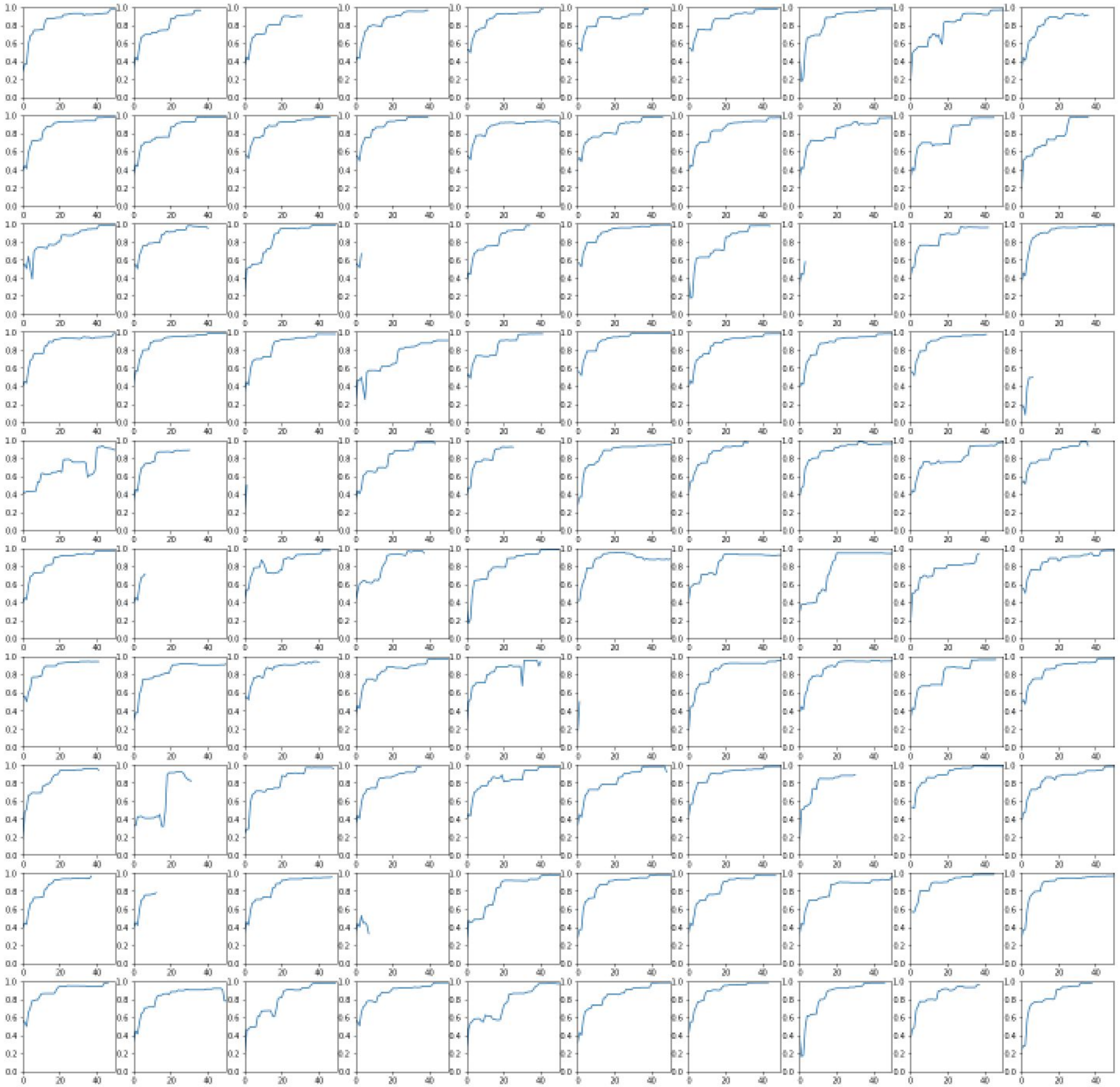
Decide When to Intervene

1. Iterate through likelihood charts to find patterns
2. Define rule based mechanism to make intervention decisions
3. Test rule performance on testing data

Decide When to Intervene

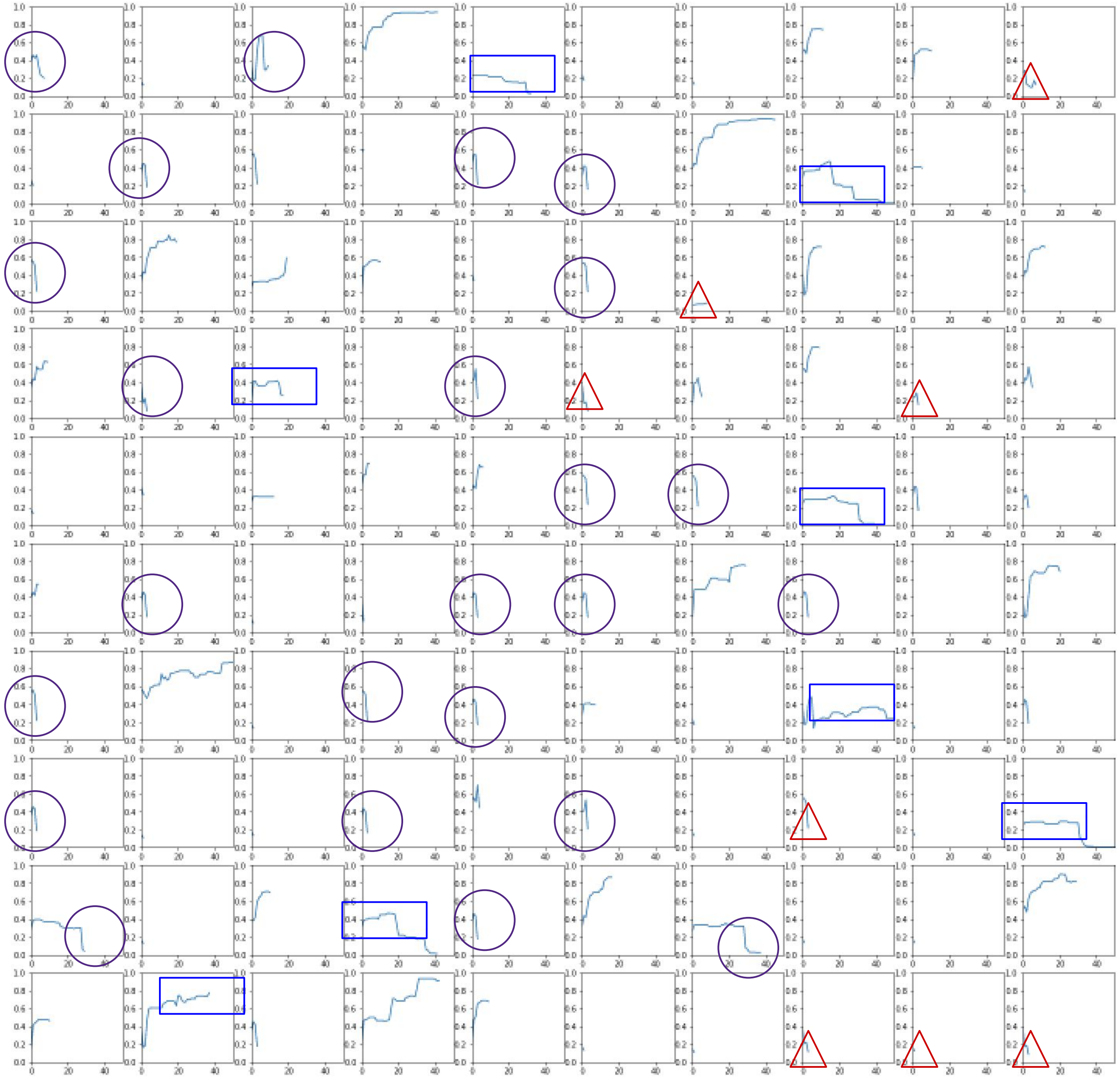
- For applicants who ended up submitting the application, their likelihood of submitting goes up constantly during the application process

- Most of them reached a likelihood of more than 70% in the first several steps



Decide When to Intervene

- Applicants who failed to submit their applications usually have a sharp drop somewhere
- They usually have trouble reaching to a very high likelihood
- It’s more likely for them to have a likelihood lower than 25%



Decide When to Intervene

Intervene when:

1. An applicant has a likelihood dropped **under 25%**
2. Continuous likelihood drop of **more than 15%**
3. An applicant has a likelihood **below 50%** after 5 events triggered
4. An applicant has a likelihood **below 70%** after 20 events triggered

Decide When to Intervene

Rule based model performance

Out of **100 predictions that a user will drop**, how many did our model **correctly** predict?
(Precision)

87

Out of **100 users dropping**, how many did our model capture?
(Recall)

71

Estimates of potential financial impact from predictive model: **Optimistic**

Intervention Matrix

Utilizing Model

		Intervention: Needed	Intervention: Not Needed
Action	**Intervention**	43,278	5,969
	No Intervention	16,722	34,031
	Total	60,000	40,000

Random Intervention

		Intervention: Needed	Intervention: Not Needed
Action	**Intervention**	36,000	24,000
	No Intervention	24,000	16,000
	Total	60,000	40,000

No Intervention

		Intervention: Needed	Intervention: Not Needed
Action	**Intervention**	0	0
	No Intervention	60,000	40,000
	Total	60,000	40,000

X

Incremental Value per Session

Intervention Needed	Intervention Not Needed
$6	-$12

=

Expected Profit

Utilizing Model:

Intervention Needed	Intervention Not Needed	Gain
$259,668	-$71,624	**$188,044**

Random Intervention:

Intervention Needed	Intervention Not Needed	Gain
$216,000	-$288,000	**-$72,000**

No Intervention:

Intervention Needed	Intervention Not Needed	Gain
$0	$0	**$0**

Assumptions:
- Successful intervention rate: **15%**
- Profit per successful intervention (app submission): **$60**
- Cost per intervention: **$3**
- Discourage Rate for Wrong Intervention: **15%**
- Total sessions (per week): **100,000**
- Dropping Rate: **60%**

Estimates of potential financial impact from predictive model: **Conservative**

Intervention Matrix

Utilizing Model

		Intervention: Needed	Intervention: Not Needed
Action	**Intervention**	43,278	5,969
	No Intervention	16,722	34,031
	Total	60,000	40,000

Random Intervention

		Intervention: Needed	Intervention: Not Needed
Action	**Intervention**	36,000	24,000
	No Intervention	24,000	16,000
	Total	60,000	40,000

No Intervention

		Intervention: Needed	Intervention: Not Needed
Action	**Intervention**	0	0
	No Intervention	60,000	40,000
	Total	60,000	40,000

X

Incremental Value per Session

Intervention Needed	Intervention Not Needed
$3	-$15

=

Expected Profit

Utilizing Model:

Intervention Needed	Intervention Not Needed	Gain
$129,834	-$89,531	**$40,303**

Random Intervention:

Intervention Needed	Intervention Not Needed	Gain
$108,000	-$360,000	**-$252,000**

No Intervention:

Intervention Needed	Intervention Not Needed	Gain
$0	$0	**$0**

Assumptions:
- Successful intervention rate: **10%**
- Profit per successful intervention (app submission): **$60**
- Cost per intervention: **$3**
- Discourage Rate for Wrong Intervention: **20%**
- Total sessions (per week): **100,000**
- Dropping Rate: **60%**

Utilizing our model,
we can target
fewer customers in
a more effective way

Assuming 100,000 users a week
we expect an incremental profit of

$188,044 per week
(optimistic scenario)

$40,303 per week
(conservative scenario)

What's next?

Opportunities For Improvements From Future Teams

1. Improve model through "Reinforcement Learning"
2. Improve model through "Recurrent Neural Networks"
3. Leverage more HEAP data

Northwestern | Kellogg

Northwestern | McCORMICK SCHOOL OF ENGINEERING

MS in Analytics