COMS E6998 Final Project: Credit Skinny

Ryan Lee DBL2127

Department of Computer Science, Columbia University

Project Resources

Project Google Doc https://docs.google.com/document/d/1GP5JTTE1KPI7ipOEQandBV6S3kjZHS edit?usp=sharing

Slide Deck Presentation https://youtu.be/P1OV7Ib8ucc

Credit Skinny Demonstration https://www.youtube.com/watch?v=UKLmM6La1sw

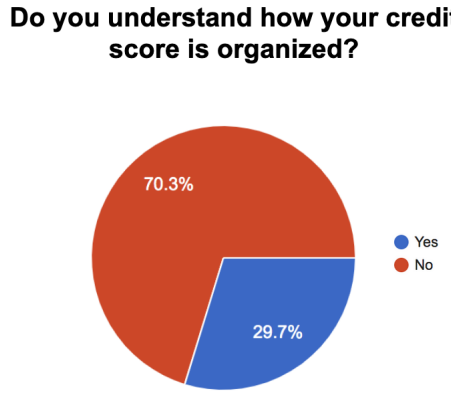
API Documentation https://app.swaggerhub.com/apis/RyanLee64/CreditSkinny/1.0.0#/

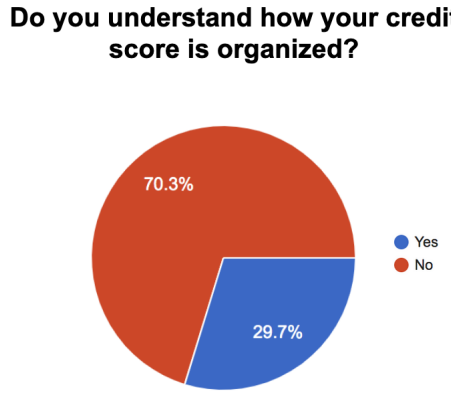
Original Credit Skinny MVP from Marketing Managment 2019 http://www.columbia.edu/~dbl2127/cloud/Credit%20Skinny%20MVP.pdf

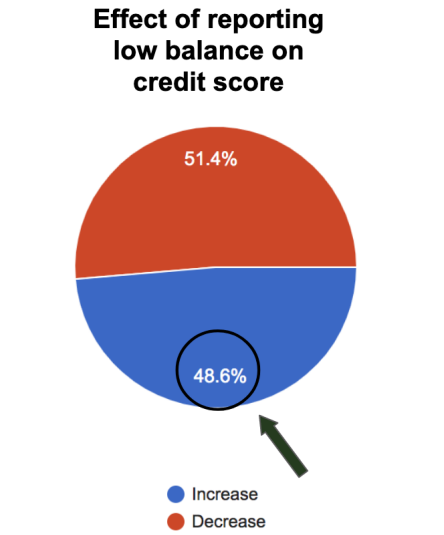
Abstract Credit Skinny is a credit education platform that intends to increase credit literacy of college aged students. It has two seminal features. The first is the on-boarding process. Within the on boarding process users specify their preferences for rewards and are then matched with three tailored recommendations that align with their preferences. In addition Credit Skinny has a module-based learning curriculum which increases credit literacy through easy to understand videos and quizzes. At the end a user of Credit Skinny should exit with an excellent credit score, three cards which are earning them great rewards, and the knowledge of how to navigate the credit landscape as they move forwards.

Introduction and Problem Statement

Credit Skinny was borne out of a 2019 project in a Columbia Course Called Marketing Management. The need for Credit Skinny arose out of two observations. The first is that Columbia students know less about credit than you would think. In a survey with forty respondents conducted in that class we found that individuals both fundamentally misunderstood very basic components of the FICO. Included are three figures drawn from the survey data conducted in that course.



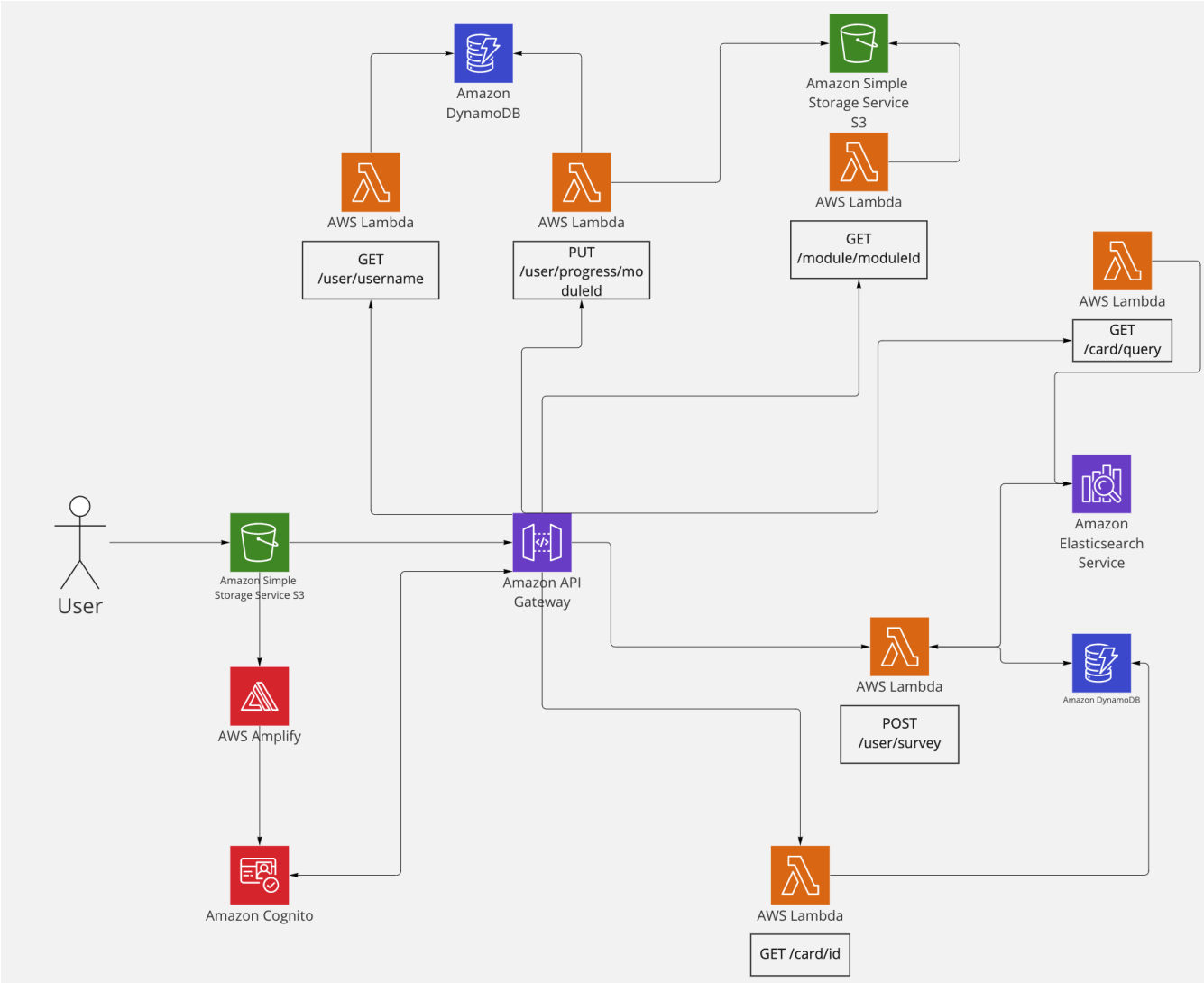




A misunderstanding of credit is an issue that extends itself beyond the gates of Columbia. In a US

Today report it was found that "The average American carries $6,000+ in revolving credit card debt, and pays upwards of $17,000+ more in interest on their mortgage alone over a 30 year stretch than someone with a credit score above 750." (https://www.usatoday.com/story/money/2019/08/18/credit-cards-report-gen-z-taking-more-debt/2008655001/). I have provided a link to the Credit Skinny MVP from that course in the resources section above.

It is my belief that this issue of rising debt and individuals ending up with cards that are sold to users rather than tailored to where they can receive the most rewards are fundamental artifacts of the current business model surrounding credit cards. Card issuers profit off of collecting interest payments, not issuing rewards. It is my belief that a module based learning curriculum interjected early into someones credit journey could have massive ripple effects down the line. Furthermore by tailoring cards to where individuals make the most money in terms of rewards we shift away from the affiliate marketing tactics which dominate credit monitoring services like The Points Guy, Credit Karma, and Nerd Wallet.



Architecture Diagram The architecture diagram for Credit Skinny is included directly above. Points of note include that the front end is hosted in an S3 bucket and written using the React framework. The userpool is managed and by Cognito. Cognito is additionally leveraged to secure the API hosted on API gateway by verifying the JWT token which users sign in an Authorization header before allowing the request to be passed through. Operation that have to do with retrieving or updating a user run through lambdas and then update a DynamoDB table of user data. Content for the module based learning curriculum resides in another S3 bucket. Content for a module consists of a .mp4 video file as a well as .json file

that contains the answers to that module's quiz questions. Survey information provided by the user after completing their onboarding process first runs a KNN routine in an Elasticsearch instance over the feature vectors derived from their survey answers and then query another DynamoDB table containing the rich human-readable data about a card.

API A brief description of the functionality of each method specified in the API can be found below.

1. post /user
   - POST method responsible for pushing a new user into the users DynamoDB table. The body of this request contains a "user" model which has fields for the first name, last name, email, current learning module, and Cognito user-id. This allows us to create the "Welcome back [name]" on the profile page as well as keep track of the user's progress through the learning curriculum and retrieve the relevant module they last left off on.

2. get /card/cardId
   - GET method responsible for retrieving a "card" model out of the "Cards" DynamoDB table. The card model is used to render the section called "The Skinny" on the front end. This contains the relevant bullet points about the card, its human-readable name, and image.

3. get /module/moduleId
   -GET method responsible for retrieving a module for a user when they are in the process of progressing through a learning curriculum. The method returns back a json formatted answer key to the current module's quiz question.

4. get /user/username
   - GET method responsible for retrieving a user's profile. The Cognito username is passed as the path parameter username. The profile allows for the frontend to get information about the user's name as well as the current module the user is on.

5. options /module/moduleId - OPTIONS method for CORS compatibility on the GET /module/moduleId method

6. post /user/uploadsurvey
   - POST method responsible for uploading the completed survey filled out by the user during the on-boarding process. This allows for the backend to generate the card recommendations which will be found in the user's "Card Vault" that they will apply to over the course of the yearlong program. These cards are tailored to the survey responses provided by the user.

7. put /user/progress/moduleId - PUT method responsible for pushing a new version of the user's profile to the DynamoDB table a of user's. This method gets called when the user successfully answers a quiz question and allows them to progress on to the next module in the learning curriculum.

8. options /user
   - OPTIONS method for CORS compatibility on the POST /user method

9. options /user/progress/moduleId
   - OPTIONS method for CORS compatibility on the PUT /user/progress/moduleId method

10. options /user/uploadsurvey
    - OPTIONS method for CORS compatibility on the POST /user/uploadsurvey method

11. options /user/username - OPTIONS method for CORS compatibility on the GET /user/username method

12. options /card/cardId
    - options method responsible for CORS on the card/cardId route

End to End Description An end to end description of the user's interaction with the application is detailed below.

1. The user arrives at the homepage and clicks "Get Started".

2. The user is navigated to a sign-up page which is backed by Cognito. Upon entering credentials a verification code is pushed to the user and they are added to the "Credit Skinny" pool of federated users hosted on AWS.

3. The user is then greeted with an on-boarding screen. The on-boarding screen sets the stage for the survey which will follow that will offer the opportunity to specify their preferences so that we can tailor card recommendations to them.

4. The user first fills out whether or not they are enrolled in college, have open lines of credit, or have a FICO score below 600. If the user is not enrolled in school or has a FICO below 600 the first card they will be suggested will be a secured card. Otherwise if they are a student without existing credit they will be recommended the Discover It Student card.

5. Next the user proceeds to fill out two additional pages about their preferences for cash back and travel rewards.

6. Upon completing the survey their responses are pushed to the backend. A "cash back" vector is created based on their responses that includes 13 dimensions. Additionally a "travel" vector is created based on their travel reward preferences. These vectors are then passed to the Elasticsearch instance. Elasticsearch instances outside of AWS have support for a field type called "dense vector" `https://www.elastic.co/guide/en/elasticsearch/reference/current/dense-vector.html`. 2x dense vectors exist for each card in the backend corresponding to the amount of points they earn in each of the 13 dimensions for either vector. A query is issued to the bakend for the travel and cash back vectors individually. The query specifies to return the KNN neighbors (in this case 1 nearest neighbor). The distance metric for KNN is cosine similarity. This idea squares well with tailoring user recommendations towards the areas of greatest relative divergence among their thirteen features. This relative notion of preference is I believe more powerful than simply trying to match 5% back on a category to a card that has exactly 5% in that category. After these queries are executed the user's "card list" is populated in the DynamoDB table.

7. The user arrives in their profile. If they navigate to their card vault they can see the three recommendations as well as details about each card presented to them as *"The Skinny"* .

8. From their profile the user can navigate to their current module. They watch a video and must successfully complete a multiple choice question before moving on to the next module.

9. Progress through the modules is displayed to the user via a status bar for them to track their progress that reflects the completion of different parts of the curriculum.

10. Finally the user may sign out from their profile and will be redirected back to the home page of the Credit Skinny site.

Divergent Profiles: DEMO FEEDBACK REQUEST

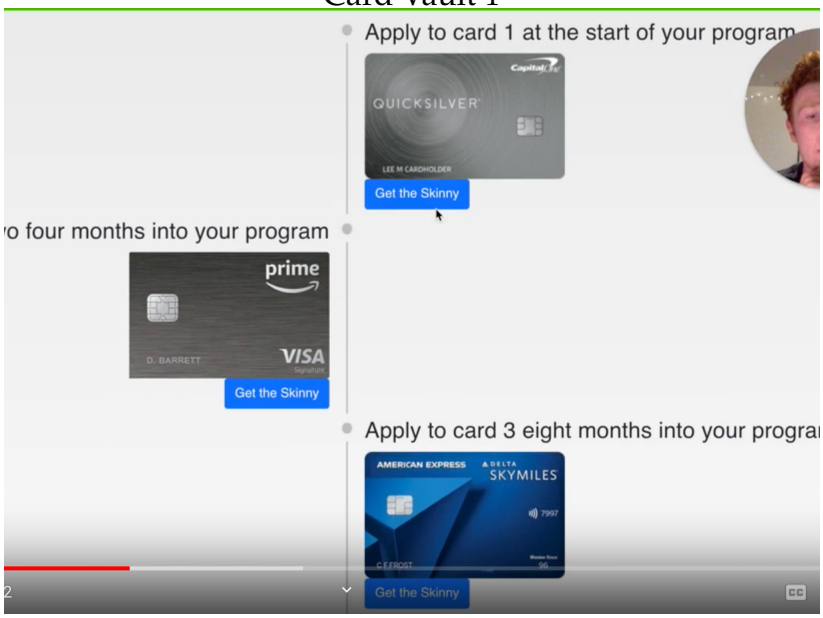
- In the demo the Professor Sahu asked that I provide an example of divergent card recommendations based on user input.

- I cover this topic at length in the video but I will summarize it below.

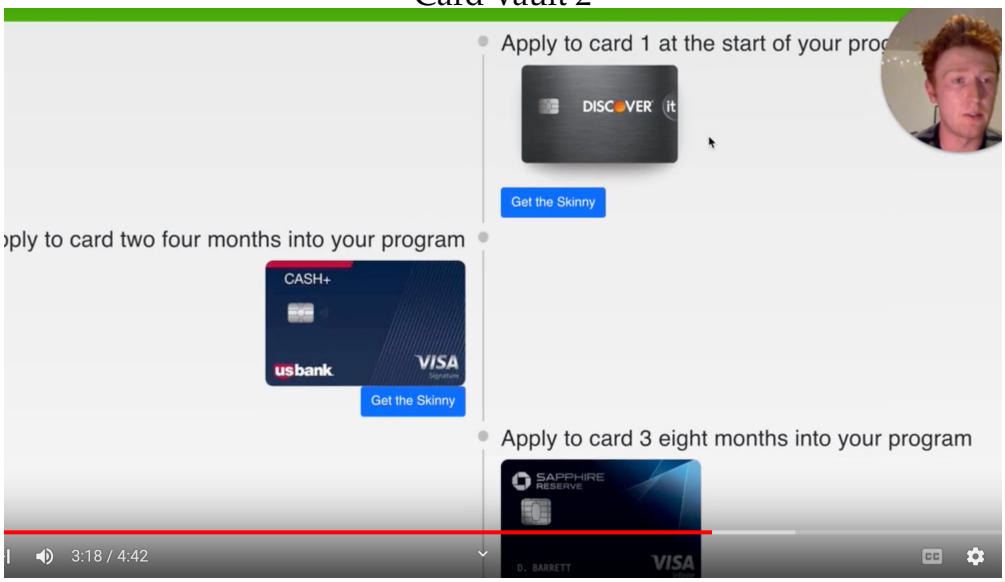
- Essentially I assume two "personas" in the demo. One is an individual that is not enrolled in college and is a brand loyalist to both Amazon for their cash back and Delta for their travel rewards. This individual cares more about having the benefits of their rewards concentrated with specific brands. This presents the upside to receive specific perks with the brands but comes at the expense of lacking flexibility when it comes to possible ways to redeem rewards.

- The second persona is an individual that is enrolled in college with no negative credit history. Instead of being a brand loyalist they care more about the breadth of their cash back rewards and travel perks that they can receive no matter the airline provider or hotel company they are staying with. I have included the "Card Vaults" generated for each of these users below.

Card Vault 1



Card Vault 2



- It is clear that neither profile is the same across the two vaults. in addition the first persona received cards tailored specifically to the brands they have a strong affinity for.
- The second user received cards that offer a breadth of categories they can earn cash back in (from grocery shopping, to gas stations, to utilities) as well as a card which offers tons of travel perks they requested such as Priority Pass, and TSA precheck.
- All of the details about the specific survey input and persona description can be found in the video demonstration of the project linked above.

Concluding Remarks I had no experience with front end development at the start of this semester. Learning React was a steep learning curve but I am thankful to be able to say that I can now develop front end code. I found out that the AWS Opensearch vector does not support the "dense vector" field type which was necessary in order to leverage the KNN routine. This required hosting the Elasticsearch instance via Elastic itself which ran on a GCP instance. Finally, I am incredibly happy with how this project came together. I had been stewing on the idea for a long time and so it was incredibly gratifying to bring it to life. I thoroughly enjoyed the course and this was a great way to conclude my time at Columbia.

Acknowledgements Many thanks to the teaching staff for all of their effort that made this semester possible. I thoroughly enjoyed the course!