# Creating a Non-Negative Matrix Factorization-Based Notes Optimization App - NoteSmart

Caroline Bromberg (Honors Capstone), Samantha Jaehnig, Varun Jindal, Michael Misiak, Ethan Morgan

University of Michigan College of Engineering, Capstone Advisor: Jeffrey Ringenberg

## INTRODUCTION

Most people organize their thoughts by taking notes. Whether it is taking organized notes for classes, making to-do lists, or keeping track of what groceries are needed throughout the week. However, once someone has accumulated a large mass of notes, each with a random to-do list or grocery item, it can be annoying to try to sift through all the notes to find the specific information that someone is looking for.

Currently, there is not a good mainstream app that can help organize these notes and perform conceptual searches through the content of the notes. With NoteSmart, people can create notes with one small item or idea on it, and the app will automatically parse the content to create relevant tags. The app can then use these tags to group notes together that have similar content (such as grouping to-do lists or shopping lists). Users can then search for particular tags, and be able to find all of their notes that have that tags in an organized manner. This app will utilize complex NLP algorithms as well as Non-Negative Matrix Factorization to parse the note content and generate tags that are relevant and helpful.

## OBJECTIVES

The primary objective of the project is to provide users with an intuitive and useful NoteSmart app that can more efficiently organize their thoughts in such a way that it is easier

for them to find the information they are looking for.

While determining what kinds of features might be the most helpful to users, I performed some research to try to answer the following questions:

- What level of combining notes is most helpful?

I wanted to figure out what level of abstraction of notes would be good for users, because too little abstraction leads to not enough grouping, and too much abstraction leads to users not knowing where to find information

- Does adding content tags make a significant difference in users' abilities to find specific information?

The main idea for the app is to implement content-based tags so that the notes can be organized in an intuitive manner, but if this does not actually help users then it will not be a good idea to implement

- Will all users find the tags intuitive, or will different users find some tags intuitive but not others?

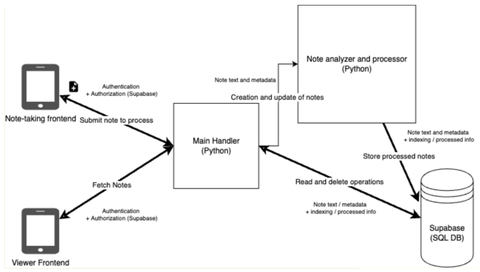
There might be many situations where some users would expect a certain tag to be associated with a certain kind of note, but a different user would expect a different tag to be associated with that note. It is important to try to create tags that are as universal as possible so that the maximum number of users will be able to understand what each tag represents in terms of their notes. Otherwise, the tag system will not be useful for many users, those users being the ones that do not intuitively understand the system of tag generation

- Does the UI feel easy and intuitive to use, or is it confusing?

A confusing UI can render an app useless, because users will not understand how to achieve their desired goals. The UI needs to be as simple as possible, while also being interesting enough to keep the user's attention without boring them. It's important to find a good balance between these things so that users want to continue to use the app

## DESIGN AND METHODOLOGY

To achieve my objectives, I used a combination of natural language processing (NLP) and matrix factorization techniques to parse note data, create relevant tags, and store them in a database. Additionally, I developed a backend and API that can send note data and receive relevant tags and create an aesthetic UI that users will enjoy using.




Engine Architecture of the App Design

NLP and Matrix Factorization Techniques:
To begin, I used existing NLP and matrix factorization techniques to parse note data and create relevant tags. I leveraged NLP algorithms to analyze the text within the notes and identify important words and phrases. I then used matrix factorization techniques to identify latent factors within the notes, such as topic clusters, which help generate more accurate and relevant tags from the data given.

Backend and API:
Another necessary component is the backend and API that can send note data and receive relevant tags. I used Python for the API calls and Supabase to store information in a database. This enabled an easy retrieval and storage of note data and relevant tags. The API calls communicated data between the app and the database, and thus could take the user-inputted notes and send it to the database, or take information from the database and run the tag-generating algorithms on it. Those tags would then be sent back to the database, and the app could display any relevant information from the database depending on which user is logged in and which notes or tags are being viewed.

Database:
To store tag and note information in a database I utilized Supabase. Supabase is an open-source backend-as-a-service (BaaS) that makes it easy to store, retrieve, and manage data. I used Supabase's database features to store note data and tags in a structured format to be easily queried and analyzed.

User Interface:
To create an aesthetic UI that users will enjoy using, I used Swift for the frontend. User-centered design principles were necessary to ensure that the UI would be intuitive and easy to use. I also leveraged existing design patterns and guidelines to create a visually appealing interface that is consistent with modern design trends.

User Testing:

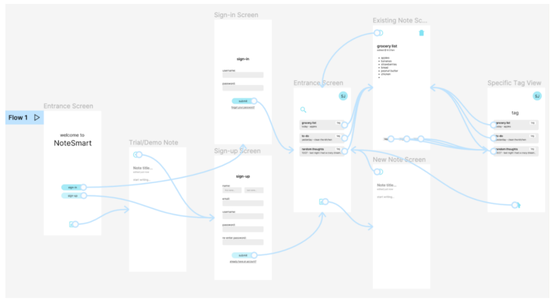
To determine the effectiveness of the UI, I performed several user tests to go through basic app tasks, such as creating notes and viewing the relevant tags, to see if there were any confusing or weak points in the UI. I then collected the data into groups to see which features needed the most reworking and which features were accomplishing all of their necessary objectives.

## RESULTS

To perform user testing on my NoteSmart note-taking app, I recruited a group of participants who had diverse backgrounds and levels of familiarity with note-taking apps. I then asked them to perform a series of tasks, such as creating a new note, finding tags, and searching for notes based on tags. During the testing, I observed their interactions with the app to take note of how each feature of the app was performing.

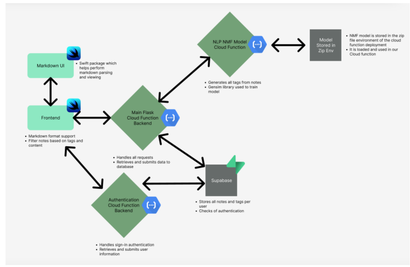
From the user testing I was able to determine that some of the features were not as intuitive to users as I wanted. For instance, a preview of note tags is shown to users on the home page, so users struggled to find a full list of all tags associated with each note, since that required extra navigating. Some participants tried to look for the list in the home page, and did not realize they needed to navigate to the notes page to expand the full list of tags.

Based on this feedback, I made several changes to the app's design and functionality to make it more user-friendly. First, I added a button to the note editor that displays all tags associated with the note when clicked, and added a signifier to tell users that the tags they were seeing were not the complete list of tags. Specifically, I added an ellipses to the end of the tag preview so users knew there was more information to view. This makes it easier for users to understand how the app works without needing to spend time exploring the app. After implementing these changes, I conducted another round of user testing to validate the effectiveness of the changes. This time, users were able to find the full list of tags more easily, and they expressed satisfaction with the new features.




Updated UI Flow

Overall, user testing was a valuable tool for identifying areas of improvement in the NotesSmart app. By observing user behavior and feedback, I was able to make changes to the app's design and functionality that improved its usability and user experience.




Updated Engine Architecture

## CONCLUSIONS

After creating a finished and reworked product of the app, and then performing user testing with that app, I was able to determine some key findings:

Based on the user testing feedback and changes made to the app's design and functionality, the tag-generation feature had a positive impact on users' overall ability to create and search for their personal notes. Users were able to navigate through their notes more quickly and find specific notes with the exact content they were looking for, even if the notes were created a long time ago. The changes made to the app's design and functionality, such as the clarification to display all tags associated with a note, improved the user experience and made it easier for users to find and organize their notes. Overall, users were able to find content in notes with an average of a 20% improved search rate between looking for specific information in notes with no tags and notes with tags.

## REFERENCES

Chawla, R. (2018, June 20). *Topic Modeling with LDA and NMF on the ABC News Headlines dataset*. Medium. https://medium.com/ml2vec/topic-modeling-is-an-unsupervised-learning-approach-to-clustering-documents-to-discover-topics-fdfbf30e27df

Egger, R., & Yu, J. (2022, May 6). *A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts*. frontiersin.org. Retrieved February 6, 2023, from https://www.frontiersin.org/articles/10.3389/fsoc.2022.886498/full

*Gensim: topic modelling for humans*. (n.d.). https://radimrehurek.com/gensim/models/nmf.html

*sklearn.decomposition.NMF*. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html

*Welcome to Nimfa — Nimfa 1.3.1 documentation*. (n.d.). https://nimfa.biolab.si/

*What are the pros and cons of LDA and NMF in topic modeling? Under what situations should we choose LDA or NMF? Is there comparison of tw. . .* (n.d.). Quora. https://www.quora.com/What-are-the-pros-and-cons-of-LDA-and-NMF-in-topic-modeling-Under-what-situations-should-we-choose-LDA-or-NMF-Is-there-comparison-of-two-techniques-in-topic-modeling

## CONTACT

Caroline Bromberg
Email: bromberc@umich.edu
Phone: (734) 730-9230
LinkedIn: https://www.linkedin.com/in/caroline-bromberg/
University of Michigan College of Engineering 2023

ENGINEERING
HONORS PROGRAM
UNIVERSITY OF MICHIGAN