Informatics Institute of Technology

In Collaboration With

University of Westminster, UK

University of Westminster, Coat of Arms

NFT-RecSys

A Trading Recommendations System for Non-fungible Tokens

A Project Specification Design and Prototype Doc by

Mr. Dinuka Ravijaya Piyadigama

w1742104 / 2018373

Supervised by

Mr. Guhanathan Poravi

March 2022

Submitted in partial fulfilment of the requirements for the

BSc (Hons) Computer Science degree at the University of Westminster.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

API Application Programming Interface.

DL Deep learning.

ERC Ethereum Request for Comments.

IDE Integrated Development Environment.

ML Machine Learning.

NFT Non-fungible Token.
NLP Natural Language Processing.

P@K Precision at K.

ABSTRACT

NFTs allow people to trace the origin of digital items and with the help of Blockchain technology. Since the items are unique from each other, as expressed by the name itself, they are *not fungible*. One NFT is expected to be unique from another. Due to several restraints that are presented with the nature of NFTs & the overwhelming amount of data that needs to be analyzed, it is difficult to find NFTs of comparable value that is trending among the community, timely and relevant to each user's identified interests or the NFT that the user currently owns.

Recommendations Systems have been identified to be one of the integral elements of driving sales in e-commerce cites. The utilization of opinion mining data extracted from trends have been attempted to improve the recommendations that can be provided by baseline methods in this research, to address the restraints presented by NFTs.

NFT-RecSys is capable of acting as a decentralized Recommendations System to provide trending recommendations of NFT assets, while preserving user-anonymity. The data extraction methods explored for recommending NFTs, integration of social-trends into recommendations & the aggregation algorithm of recommendations from ensembled models are novel results yielded by this research.

Keywords: Recommendation Systems, Hybrid Recommendation Systems, Machine Learning, Non-fungible Tokens, Data Science, Opinion Mining

Subject Descriptors: Natural Language Processing (I.2.7), Distributed Systems (C.2.4), Distributed Artificial Intelligence (I.2.11), Electronic Commerce (K.4.4), Social an Behavioral Sciences (J.4), Algorithms (I.1.2)

CHAPTER 1: INTRODUCTION

1.1 Chapter Overview

In this research project, the author tries to identify the required features to be considered for an NFT-trading Recommendations System and introduce a new Ensemble Architecture for Recommendations that can be applied in other related domains as well. The proposed architecture will try to automate several decision-making steps that a user would otherwise need to go through to find the best possible trade.

This chapter defines the problem, the research gap, aims & objectives of the research and the research challenge that the author wishes to address by completion of the project. The necessary proofs of the problem, as well as previous research interests, are also reviewed.

1.2 Problem Domain

1.2.1 Non-fungible Tokens (NFTs)

In recent months, the NFT market has been growing exponentially as it appears to be the most widely accepted business application of Blockchain technology (Dowling, 2021b), since the introduction of crypto. NFTs are provably scarce unique digital assets that can be used to represent ownership (*ERC-721 Non-Fungible Token Standard* 2021). They can be one of a kind rare artworks, collectable trading cards, and other assets with the potential to increase in value due to scarcity (Conti, 2021; Fairfield, 2021). While being digital assets, they also can be used to represent physical assets. A digital certificate of land/ qualification can be identified as a couple of examples. The biggest winners in the NFT space over the last few months have been digital artists who were able to sell art worth over $2.5 Billion (*Off the chain* 2021).

NFTs were introduced by Ethereum (Wood, 2014) as an improvement proposal (*EIP-2309* 2021; *ERC* 2021) in the Ethereum Request for Comments (ERC)-721 standard (*ERC-721 Non-Fungible Token Standard* 2021). This allows anyone to implement a Smart Contract with the ERC-721 standard and let people mint NFTs as well as, keep track of the tokens produced by it. This allows the created tokens to be validated.

Each of these created tokens is unique from the other tokens created by the same Smart Contract, unlike fungible tokens which were introduced with cryptocurrencies and are denoted by the ERC-20 standard (*ERC-20 Token Standard* 2021) on the Ethereum network. One Bitcoin can be swapped to another Bitcoin, but each NFT will be unique. Then, the deployed Smart Contract will be responsible to keep track of the tokens created by it on the network. A Smart

Contract is a program that resides on the Ethereum network with a collection of code & data (*Introduction to smart contracts* 2021).

For each NFT, the contact address & unit256 tokenId are globally unique on any blockchain. This allows Decentralized Applications (DApps) (Frankenfield, 2021; *Decentralized applications (dapps)* 2021) to take the tokenId and present the image/ asset that is identified by the particular NFT.

"To put it in terms of physical art collecting: anyone can buy a Monet print. But only one person can own the original." (Clark, 2021)

While a digital file can be copied regardless of whether it's an NFT or not, what this technology provides is the ownership of the digital asset. If an NFT that contains your certificate/ domain is held under your wallet on the Blockchain, no one else can get it from you unless they have your digital wallet's private key. Similar to a deed. But, anyone can see, validate and admire what you own.

1.2.2 NFT Marketplaces

OpenSea, which was the first NFT marketplace is also considered to be the largest. In the attempt to become the "Amazon of NFTs", OepnSea raised $23 million in a Series A (Hackett, 2021), following a $100 million raise in a Series B round, ended the company in a valuation of $1.5 billion (dfinzer, 2021; Matney, 2021). Open Sea saw nearly $150 million in sales in the month of June. These marketplaces are set to increase access to the digital goods industry (Chevet, 2018).

An NFT purchased on an Ethereum marketplace can be traded on any other Ethereum marketplace for a completely different NFT. Creators don't necessarily need to sell their NFT on a market. They can do the transaction peer-to-peer, completely secured by Blockchain. No one is needed to intermediate and an owner isn't locked onto any platform (*ERC-721 Non-Fungible Token Standard* 2021).

1.2.3 Recommendation Systems

Recommendation Systems have been driving engagement and consumption of content as well as items on almost every corner of the internet over the last decade. These systems help users identify relevant items on an online platform. When users are recommended with relevant items, it enables businesses in growing their revenue. 35% of Amazon's revenue (Naumov et al., 2019) & 60% of watch time on YouTube (*Recommendations* 2021) comes from recommendations.

1.3 Problem Definition

Currently, there is no way of identifying possible tradable NFT assets, unless manually browsing through the internet. Marketplaces allow searching for NFTs by keywords, categories & pricing, but don't provide personalized recommendations of trending items. This applies to someone who wants to purchase an NFT that shows similar characteristics to another NFT that has already been purchased by a previous buyer or oneself. Since there can be only one owner for an NFT at a time, recommendations using standard collaborative filtering is also not entirely ideal. Content-based approaches won't help identify trending items.

To help with the exploration of these digital assets, it's identified that several steps that the user has to follow to identify trending items that are timely, popular among the community and may have an expected value can be automated.

1.4 Research Aims and Objectives

1.4.1 Research Aim

The aim of this research is to design, develop & evaluate a novel Recommendation Architecture that will provide relevant, trending, timely, and worthy NFTs for trading purposes by automating some of the decision making steps that the user would otherwise have to do manually.

To elaborate on the aim, this research project will produce a system & architecture that can be used to recommend trending items with respect to a chosen item in a specific data set. The focus will be laid on the recommendation of NFTs. In order to achieve this several public channels of trends will be required to be streamed into the recommendations architecture together with the automation of several decision-making steps that a user that is interested in purchasing NFTs would have to manually go through, in order to make the best possible trade. The use of Data Mining techniques, Natural Language Processing (NLP) techniques, Data Analysis, hybrid, content-based, collaborative filtering & Deep Learning methods will be researched to make the best possible recommendations.

The required knowledge will be studied and researched, components will be developed and the performance will be evaluated in order to validate or invalidate the chosen hypothesis. The system will be able to run in a local browser for personal use or in a hosted server for public use. The data science models & their code will be available for further research and use in a public repository that is easy to get up and running with ease. A review paper will be published with knowledge gathered from the survey of Literature. A research paper will be published on the outcome of the findings in the research project.

1.4.2 Research Objectives

The Aims and Research Questions mentioned above are expected to be achieved and answered with the completion of the following Research Objectives. These objectives are milestones that will be expected to be met in order for the research to be completed successfully.

Table 1.1: Research Objectives

Objective	Description	Learning Outcomes	RQ
Literature Survey	Read previous work to collate relevant information on related work and critically evaluate them. • **RO1:** Conduct a preliminary study on existing Recommendations Systems & Architectures. • **RO2:** Analyze the perception of Recommendation techniques. • **RO3:** Conduct a preliminary study on NFTs. • **RO4:** Analyze user desires and factors that affect the likability of owning NFTs.	LO4, LO2, LO5	RQ1 RQ3
Requirement Analysis	Specifying the requirements of the project using appropriate techniques and tools in order to meet the expected research gaps & challenges to be addressed based on previous related research and any domain-specific sources of knowledge. • **RO5:** Gather information about requirements related to desirability of owning NFTs & crypto-related assets. • **RO6:** Gather the requirements of a Recommendations System and understand end-user expectations. • **RO7:** Get insights & opinions from technology & domain experts to build a suitable system.	LO1, LO2, LO5, LO7	RQ1 RQ2 RQ3

Design	Designing architecture and a system that is capable of solving the identified problems with recommended techniques. • **RO8:** Design a price prediction system to identify the possible increase/ decrease in value of the NFTs. • **RO9:** Design an automated flow to match NFTs with global social trends data. • **RO10:** Design a data-preprocessing pipeline to add Smart Contract data related to NFTs in the system. • **RO11:** Design a Deep learning (DL) or Machine Learning (ML) Recommendations model that is capable of appropriately utilizing feature-enhanced data to produce recommendations.	LO1	RQ2 RQ3
Development	Implementing a system that is capable of addressing the gaps that were aimed to be solved. • **RO12:** Develop a Recommendations System that can produce relevant, timely & trending NFTs (items). • **RO13:** Integrate automation steps in the prototype to enhance features of NFT records and use them to recommend suitable NFTs. • **RO14:** Develop an algorithm that can utilize factors that are considered to affect the desirability of owning an NFT by a person.	LO1, LO5, LO6	RQ1 RQ2 RQ3
Testing and Evaluation	Testing the created system & Data science models with appropriate data and evaluating them with baseline techniques identified in the literature. • **RO15:** Create a test plan and perform unit, integration and functional testing. • **RO16:** Evaluate the novel model by bench-marking with Precision at K (P@K) score, compared against baseline models.	LO4	RQ1 RQ2 RQ3

Documenting the progress of the research	Documenting and notifying the continuous progress of the research project and any faced obstacles.	LO8, LO6	RQ1 RQ2
Publish Findings	Produce well-structured documentation/ reports/ papers that critically evaluate the research. • **RO17:** Publishing a review paper on related work. • **RO18:** Publishing evaluation & testing results identified from the research. • **RO19:** Making the code or models created in the research process available for future advancements in research. • **RO20:** Making any modified data-sets or re-creation strategies available to the public, to train & test models related to similar use cases of utilized data.	LO4, LO8	RQ1 RQ2 RQ3

1.5 Novelty of the Research

The author's research contribution that highlights the novelty of the research can be identified as follows:

1.5.1 Technological Novelty

A Hybrid Recommendations technique that attempts to use public trends in a way that hasn't been attempted in previous research will be explored in order to facilitate the recommendation of relevant, trending and timely items. Automation of several decision-making steps that a user would otherwise need to go through to find the best possible trade will be integrated into the Recommendations Architecture. It is hypothesized that this novel recommendations architecture will be able to be applied to other items as well to give enhanced recommendations based on trends.

1.5.2 Application Novelty in Domain

Currently there is no research work done regarding the recommendation of NFT assets. The information in an NFT that has an effect on a user's desire to be owned will be identified, when attempting to provide suitable recommendations. Looking at the success of Recommendation Systems across multiple systems for over a decade, it is understood that a Recommendation

System would help users identify NFTs that they would be interested in trading. This will in return help in increasing sales on NFT Marketplaces and wider adoption of the technology.

1.6 Research Challenge

NFTs is a new domain, which has very less research done related to preferences and factors considered when purchasing NFTs. Therefore, it is first important to identify the data points (features) & external factors that affect the value/ desirability of owning NFTs to suggest trading recommendations of NFTs to a user.

"Crypto has a founding tradition of emphasizing freedom and privacy. Maybe because of this prevailing cultural trend, the NFT space does not have many recommender systems." (Theorem, 2020)

NFTs are identified to be more challenging to be recommended to users using traditional recommendation methods due to the uniqueness of each item together with the traditions brought forward with the crypto community. Similar to cryptocurrencies, it has been identified that NFTs too have an impact on the general public opinion & trends (Dowling, 2021a).

Currently, available Recommendation Systems haven't had the necessity to consider trends as much as with related to the desirability of owning NFTs. Furthermore, scarcity of items opens another challenge of the inability to keep recommending items that are not available for sale or have already been purchased by an interested buyer. But that alone can't be considered due to the time-tested & proven baseline recommendation techniques being highly effective in multiple domains. Using the identified factors to be considered, a suitable recommendations architecture needs to be implemented.

1.7 Chapter Summary

This chapter presented the problem with necessary proofs and domain description, the research gap, the research challenge, and the research strategy that is expected to be addressed by the author in the research project presented by this document. The research objectives were mapped to the learning outcomes of the project module in the BSc(Hons) Computer Science undergraduate program of the University of Westminster.

CHAPTER 2: SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Chapter Overview

This chapter focuses on identifying possible stakeholders of the project by taking a look at all possible points of interaction with the system with the use of a rich picture diagram, gathering their perceptions to analyse and come up with possible expected use cases, functional and non-functional requirements of the prototype.

2.2 Rich Picture

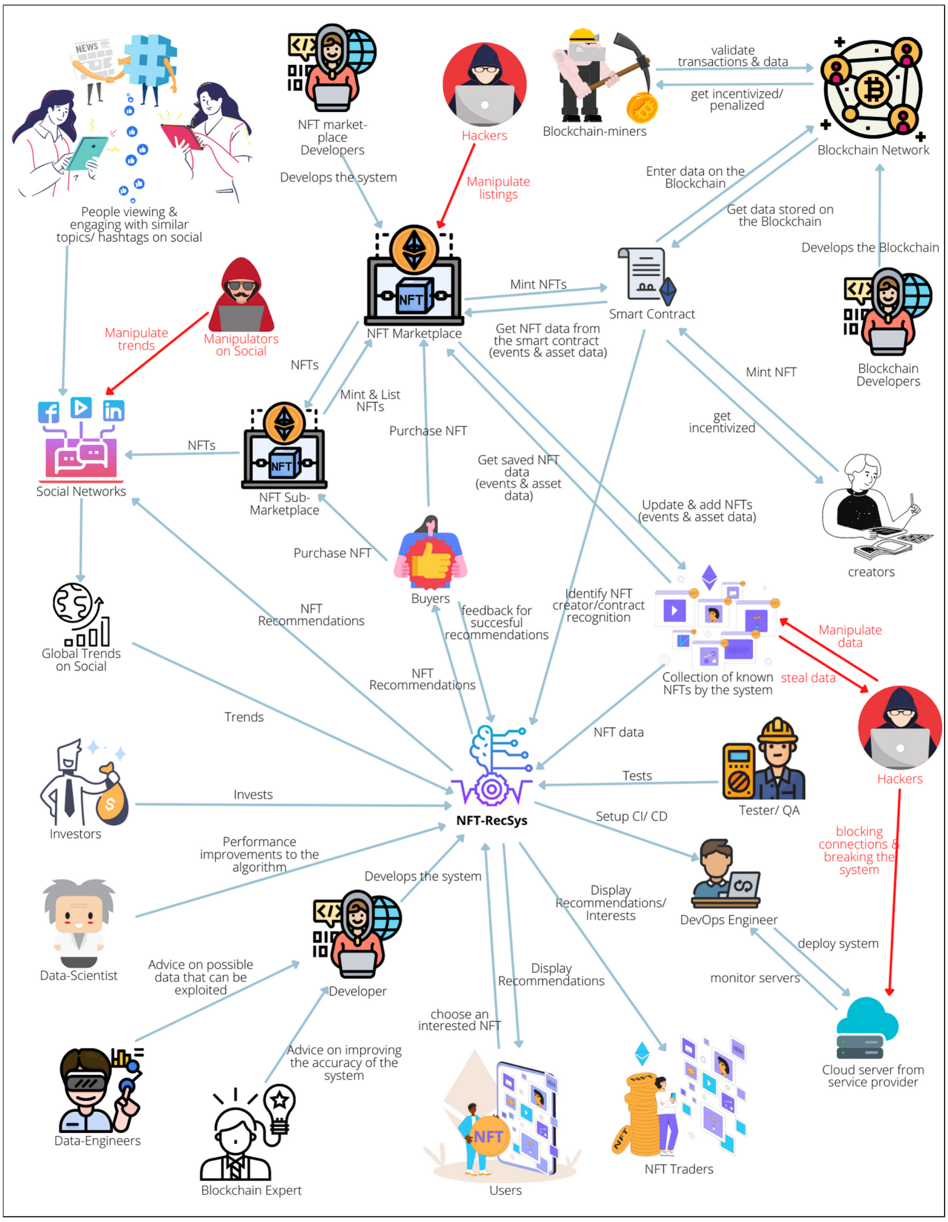


Figure 2.1: Rich Picture Diagram *(self-composed)*

The above Rich Picture diagram shows a helicopter view of how related parties in the rest of the world interacts with the system. It is used to understand the possible interactions that are expected to happen when the system is functional.

2.3 Stakeholder Analysis

The Stakeholder Onion Model illustrates recognized stakeholders who are associated with the system, along with an explanation of each stakeholder's involvement in the system, in Stakeholder Viewpoints.

2.3.1 Stakeholder Onion Model

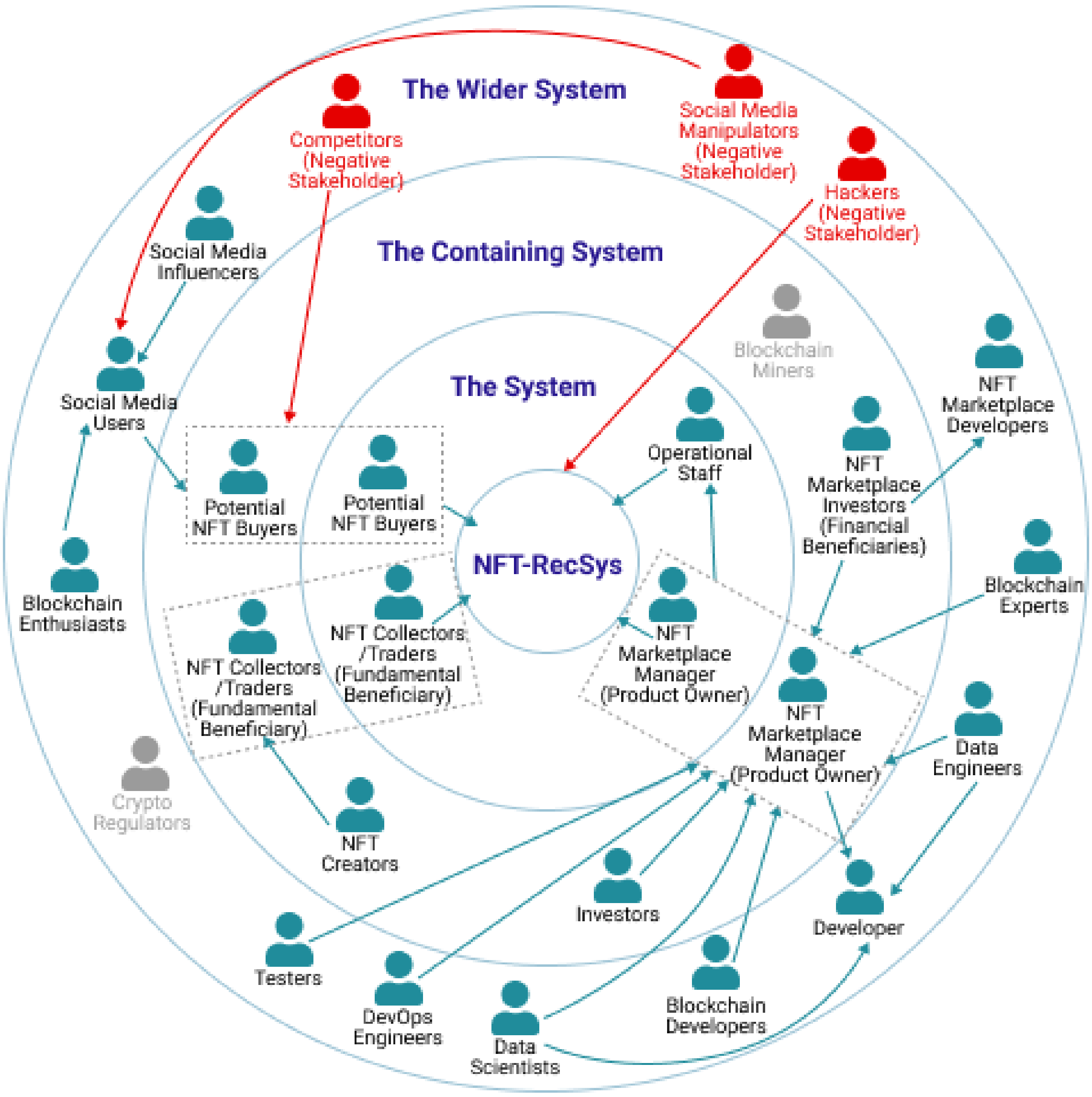


Figure 2.2: Stakeholder Onion Model *(self-composed)*

2.3.2 Stakeholder Viewpoints

Table 2.1: Roles and benefits of identified stakeholders

Stakeholder	**Role**	**Benefits/ Role Description**
Developer	Financial Beneficiary	Develops the system
Investors		Makes a profit out of the investments put into marketing, deployments and development of the system
NFT Marketplace Developers	Operational - Maintenance	Integrates the system into NFT Marketplaces.
Blockchain Experts	Expert, Quality Regulator	Provides expert advice & insights into domain knowledge, to improve the system's performance.
Data Scientists		Provides performance improvements for the performance of the Data scienc models/ algorithms used.
Data Engineers		Provides advice on possible data that can be exploited, to make the best possible recommendations.
NFT Creators	Financial Beneficiary	Gets a better opportunity to get their creations in the eye of potential buyers. Makes a profit by selling creations to people who are interested in the creations.
NFT Traders/ Collectors	Fundamental Beneficiary	It becomes easier for traders to sell NFTs as well as explore more NFTs to purchase. It also allows them to explore NFTs that may be worth collecting for a future trade.
Potential NFT Buyers		It becomes more convenient for these parties to explore NFTs that they're interested in.
NFT Marketplace Manager	System Owner, Operational - Administration	Inputs data sources for opinion mining, sets default biases. Makes sure that the system is up & running, while managing the operational staff.

Operational Staff	Operational - Support	Makes sure that the system is up & running, while attending to users' requests & issues.
DevOps Engineers	Product Deployment & Maintenance	Deploys the system to the cloud and make sure that it's up & serving users, without throttling.
Social Media Influencers	Operational - Secondary	Influences users on social media and drives trends.
Social Media Users	Operational - Secondary & Fundamental Beneficiary	Get influenced to search for items of interest and possibly turn into potential NFT buyers.
Hackers	Negative Stakeholder	May manipulate listings in NFT market places.
Competitors		May build competing products that outperform/ undercut pricing.
Social Media Manipulators		May manipulate users on social media & drive trends that a majority of users aren't interested in.
Blockchain Enthusiasts	Operational	Helps drive awareness and keep the public up to date with the latest releases & feature updates.
Blockchain Miners	Operational - Secondary	Helps keep Blockchains up & running by validating the data on the network.
Crypto Regulators	Quality Regulator	May have an impact as a regulator, if the system is used by mainstream networks.
Testers	Quality Inspector	Tests the system & ensures that it's suitable to run in production.

2.4 Requirement Elicitation Methodologies

In order to gather requirements for the development of the research project, there were multiple requirement elicitation methodologies that were followed. literature review, interviews, survey & prototyping were the methodologies chosen for this purpose. The reasons to choosing the specified requirement elicitation methodologies have been discussed below.

Table 2.2: Requirement Elicitation Methodologies

Method 1: Literature Review
At the inception of the project, the author has done a thorough literature review to identify research gaps that are open in the desired field of study and a chosen domain of interest. In order to understand research gaps available in technologies that can be applied, existing systems were studied together with relatable technologies that are possible to be applied to the existing systems that were mentioned in literature.
Method 2: Interviews
Interviews were conducted as a means of gathering expert-insights into domain-specific requirements and also to identify the best possible way to solve the problem at hand while contributing to the body of knowledge through research. Due to the domain being new and the required technical knowledge being specific, interviews were identified to be the best-possible source of knowledge to gather requirements that align with the research gap. This method also allowed to get qualitative feedback on the proposed system making it possible to identify any drawbacks/ challengers that may have to be addressed while prototyping.
Method 3: Survey
As a means of conducting a survey, questionnaire was used as a tool to gather requirements and insights from potential users of the proposed system. This form of survey will aid the author in comprehending people's cognitive processes and the expectations they have for the prototype. It will also allow the author to clarify if the proposed solution would be helpful to intended users.
Method 4: Prototyping
Since the project was chosen to follow the *Agile* Software Development Life-cycle, prototyping would allow the author to recursively try out various alternative implementations to identify any areas of improvement while testing and evaluating the prototype.

2.5 Analysis of Data & Presentation of the Outcome through Elicitation Methodologies

The analysis of data gathered through the chosen means of requirement elicitation have been presented below.

2.5.1 Literature Review

Table 2.3: Findings through Literature Review

Finding	**Citation**
In completion of the review of literature, it was identified that a Recommendations System for NFTs would benefit the majority of users to make purchase decisions as well as allow them to explore relevant items, that would in return benefit the market places, creators & traders who are selling them as Recommendations Systems have proven to improve sales of e-commerce sites in the past.	(Naumov et al., 2019; Vanderbilt, 2021)
When exploring technologies that can be applied to achieve the required outcome, it was understood that the use of Deep learning hasn't been able to improve the output of recommendations compared to other fields of applications, in most cases.	(Choi et al., 2021)
It was identified that implementing a custom hybrid ensembled model with the injection of social media trends has not been explored in literature.	(Ayushi and Prasad, 2018; Cheng and Lin, 2020)
The use of data from similar users' timelines for recommendations has been mentioned as possible future work.	(Chen and Hendry, 2019)
Pricing of NFTs & contract recognition data have not been considered for any previous implementations of Recommender Systems	(Theorem, 2020)
The only study related to recommending NFTs only recommends NFT collections that a user may be interested in, but not actual NFTs themselves.	(ibid.)

2.5.2 Interviews

In order to get opinions of technical as well as domain expertise, interviews were conducted with experts from the respective fields. Experts & researchers in ML, Recommendation Systems and Blockchain were chosen to be interviewed in order to establish project requirements. 3 Blockchain experts, 1 NFT Creator, 1 Senior Data Engineer, 2 PhD students in ML and a Data science engineer were interviewed. The outcome of interviews were processed to a **thematic analysis** based on the following themes.

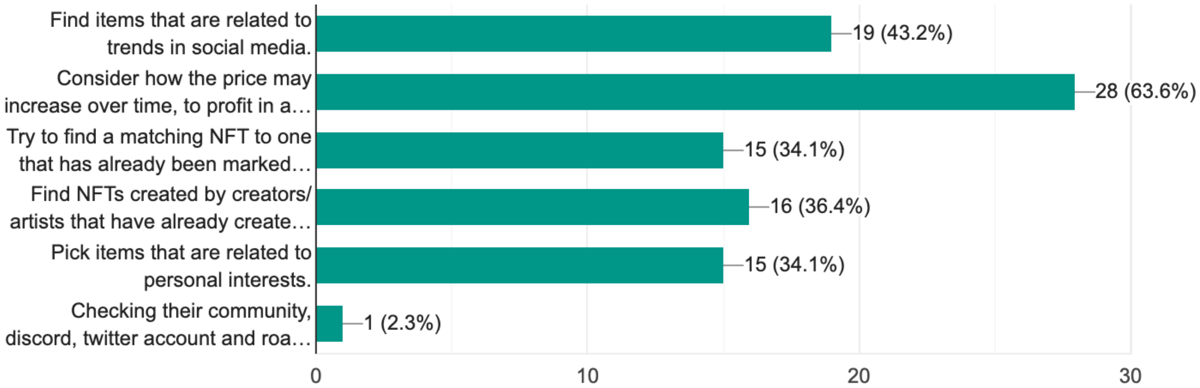
Table 2.4: Thematic analysis of interview findings

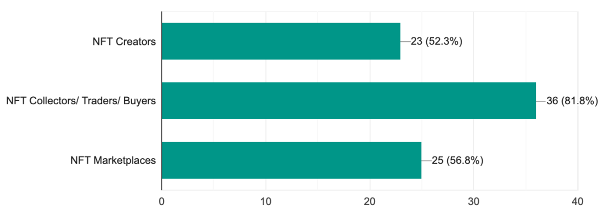
Theme	Analysis
Collection & pre-processing of available data.	As this is expected to be a Data science project, the main concern that all participants had was the availability of data. Clustering of available data was suggested to identify possible patterns by ML experts, while Blockchain experts suggested the use of publicly available data on the Blockchain such as details from Smart-Contracts to be used to improve the quality of recommendations.
Applicable Recommendation Techniques	The opinion of majority of the interviewees was that this project would benefit more by the use of rule-based algorithmic recommendation models instead of DL models due to the constraint of . According to technical experts, having a specialized recommendation model built using algorithms is very highly accepted in industrial applications. They seem to perform better in most new domains according to PhD researches. Even some of the biggest e-commerce organizations in the world seem to benefit a lot by custom-built recommendations algorithms tailored to specified use-cases according to research & development experts in Recommendation Systems.
Integration of Opinion Mining into Recommendation Systems	Domain experts thought that integrating trends and other social opinion will add value to the recommendations. They were also interested in identifying a possibility of checking for the sentiment represented by the opinions as well. When considering social sentiment, Tweets/ opinions of well-known influencers may play a bigger effect into the value of curtain NFTs.
Research gap & scope	The technological experts thought that the method that the author proposed was very innovative and that according to their knowledge, they haven't seen a similar integration to the suggested architecture in previous applications.

Creating the bias for a Hybrid Recommendations Model	While some of the interviewees suggested the use of a fixed weighted bias, others suggested a variable bias. The method applicable for variable bias or the best-possible fixed bias can be tested via continuous prototyping & evaluation. The use of user-input was also suggested to identify a possible expected bias.
Prototype features & suggestions	The Data science experts were very interested in seeing a Recommendations System built purely using custom algorithms with the help of vectorization functions that many ML libraries support. The use of transfer learning or pre-trained models were suggested for NLP parts of the implementation.
Understanding a buyer's decision making for automation	The value proposition was identified to be created by an external entity based on contract & token Ids stored on the blockchain. Due to the difference in real world trust and blockchain trust, this may have to be inferred from the available data such as past contract data and social sentiment from trends.
The necessity of NFT-RecSys & contributions	As the first research study related to a Recommendations System for NFTs, the interviewees thought that the contribution to the domain will be of great value and also, since the hybrid architecture of the proposed system is novel, the contribution to the technological domain would help the advancement of the quality of recommendations in future implementations. It was also understood that it's difficult to find specific NFTs based on tags/ characteristics. Furthermore, it was revealed that Sri Lanka does not have Machine Intelligence/ Data science driven Recommendation Systems in all local e-commerce stores.

2.5.3 Survey

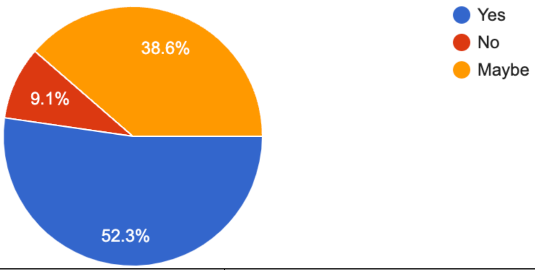
Table 2.5: Analysis of replies to questionnaire

<table>
<tr><td>Question</td><td>How will you decide which NFT to purchase?</td></tr>
<tr><td>Aim of question</td><td>To understand how a potential buyer would proceed to purchase an NFT.</td></tr>
<tr><td colspan="2">Findings & Conclusion


A majority of the participants thought that considering the price increase over time would be the primary factor of consideration when purchasing an NFT, while the second most impact to be considered was trends in social media. Finding NFTs that have been created by creators/ artists who have created valuable NFTs in the past, an NFT that is similar to what is already highly valuable and picking items related to personal interests saw similar weightings when making purchase decisions.</td></tr>
<tr><td>Question</td><td>Who do you think will be benefited from using this system?</td></tr>
<tr><td>Aim of question</td><td>To identify the beneficiaries of the proposed system.</td></tr>
<tr><td colspan="2">Findings & Conclusions


While more than 50% of participants aggreed that the proposed system would benefit the suggested beneficiaries, 81.8% thought that NFT collectors/ traders/ buyers would benefit. Since, they are the ultimate target users, it's satisfying to see such positive responses.</td></tr>
<tr><td>Question</td><td>Do you think that this system would benefit people who have no expertise in Blockchain/ NFTs as well as people who have a decent amount of expertise in Blockchain/ NFTs?</td></tr>
<tr><td>Aim of question</td><td>To identify how valuable the system would be to people of all levels of expertise in Blockchain/ NFTs</td></tr>
</table>

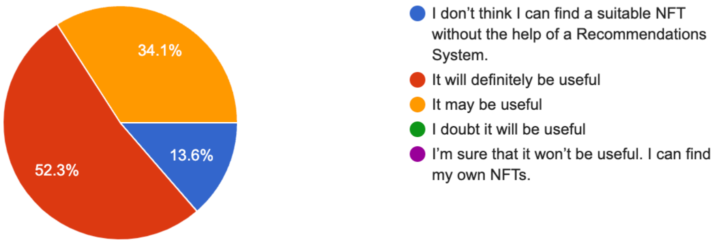
Findings & Conclusion



With majority of the responses suggesting that people of all levels of expertise in Blockchain/ NFTs would benefit from the system depicts that the proposed system would be beneficial for above-average users as well.

Question	**How much do you think that a Recommendations System would benefit you, if you ever plan on purchasing an NFT?**
Aim of question	To identify if the respondents think that the system would benefit them.

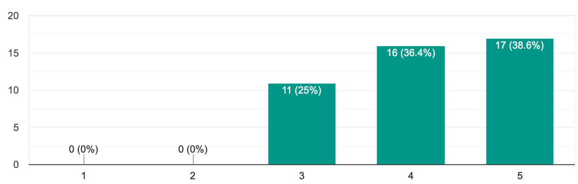
Findings & Conclusion



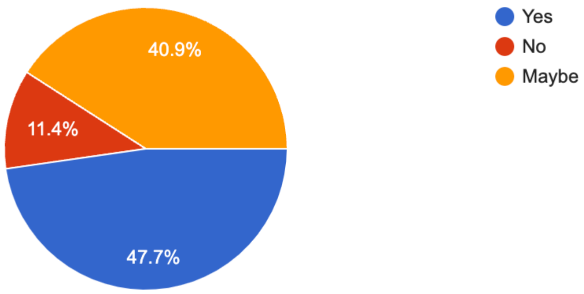
52.3% of users thought that a Recommendations System would definitely be useful to them if they plan on purchasing an NFT, while 34.1% thought that it may be useful. Meanwhile, 13.6% of users thought that they don't think that they could find a suitable NFT without the help of a Recommendations System. 100% of the results were aligned towards seeing a possible benefit of the proposed system.

Question	**How much would you expect a Recommendations System that considers social media trends to be beneficial for businesses to integrate into their online platforms?**
Aim of question	To identify the importance of the technological contribution in the project

Findings & Conclusion



The results from this question suggests that the technological contribution that has been highlighted in this project, which addresses an advancement of development of Recommendation Systems is expected to be extremely beneficial for business applications.

<table>
<tr><td>Question</td><td>Do you think that a user would benefit more if one platform provides recommendations that differ from another platform with the same dataset?</td></tr>
<tr><td>Aim of question</td><td>To identify if the proposed Recommendations System will benefit from implementing a Reinforcement Learning technique or a variable bias to adapt and suite different platforms.</td></tr>
<tr><td colspan="2">Findings & Conclusion


A majority of participants thought that having varied recommendations in different platforms, using the same recommendations algorithm. This leads to the requirement of implementing a variable bias towards the factors considered for recommendations or implementing a reinforcement learning technique, for the model to adjust based on user-inputs. Having a pre-configurable bias will also allow to achieve this, but the results from recommendations may not be optimum.</td></tr>
<tr><td>Question</td><td>What functionalities would you like to have in a Trading Recommendations System for Non-fungible Tokens?</td></tr>
<tr><td>Aim of question</td><td>To identify the non-function requirements of the system, that would make the system as user-friendly as possible</td></tr>
<tr><td colspan="2">Findings & Conclusion
Most responses form the participants revolved around considering price-predictions when making recommendations. There were also suggestions to integrate trending crypto news to the system. Suggesting potential NFTs that suit a person's personal interests were also suggested to be integrated.</td></tr>
</table>

2.5.4 Prototyping

Through iterative prototyping, there were many requirements & challengers that emerged. Firstly, there was no dataset. The data had to be pulled from an open Application Programming Interface (API) and filtered. The main challenge that was met here was the overwhelming amount of data that was received related to each NFT and rate limits of the API. The data received had to be filtered quite a lot and the most usable data points possible to be used for recommendations had to be identified & extracted. Not all NFTs contained usable content-

information. This had to be addressed with normalizing several fields and finding alternatives to map items using other available data.
The integration of social trends data brought in a new valid perspective that could be used for recommendations.

2.6 Summary of Findings

Table 2.6: Summary of Findings

Id	Finding	Literature Review	Interviews	Survey	Prototyping
1	The proposed system would benefit experienced & inexperienced users searching for NFTs as well as NFT creators, traders & market places	✓	✓	✓	
2	The limits of Recommendation Systems can be pushed without the use of Deep learning, by the application of various hybrid ensemble models	✓	✓		
3	The integration of social media trends would be beneficial to improve recommendations produced by a Recommendations System	✓	✓	✓	✓
4	The identified research gap would contribute to both the Blockchain-NFT domain as well as the advancement of Recommendations Systems & ML	✓	✓	✓	
5	Building custom use-case specific algorithms for the Recommendations System is prefered over the use of pre-built models from a business application perspective		✓		
6	Having a method of price-prediction & using the prediction data to make decisions on recommendations would benefit users		✓	✓	
7	Using data-clustering techniques to identify contract-recognition & data tags are expected by advanced-users		✓		
8	Personalized recommendations could be achieved by the use of information extracted from the Blockchain with related to a user's public key. Past purchases of NFTs made by users can be considered.	✓	✓		

9	It would be good to have a user-interface that allows the user to choose the bias/ his primary concerns when expecting a recommendation, to provide the perfect recommendation for each user.		✓		
9	Having a adaptable, variable Recommendations Model that allows different platforms to have varied recommendations is preferred.		✓	✓	
10	Having a sufficient set of well-cleaned & pre-processed data would be vital for the performance of the system	✓	✓		✓
11	Opinions of well-known influencers could have a bigger impact on the decision-making process of a majority of users.		✓		

2.7 Context Diagram

Prior to development, the system's boundaries and interactions should be determined. The system's context is depicted in the diagram below.

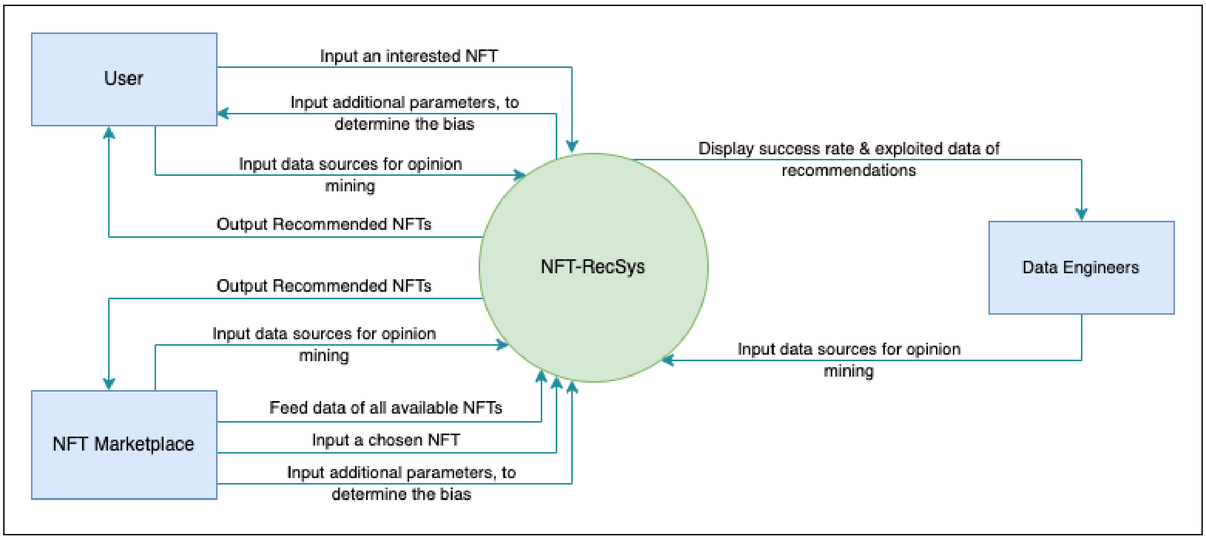


Figure 2.3: Context Diagram *(self-composed)*

2.8 Use Case Diagram

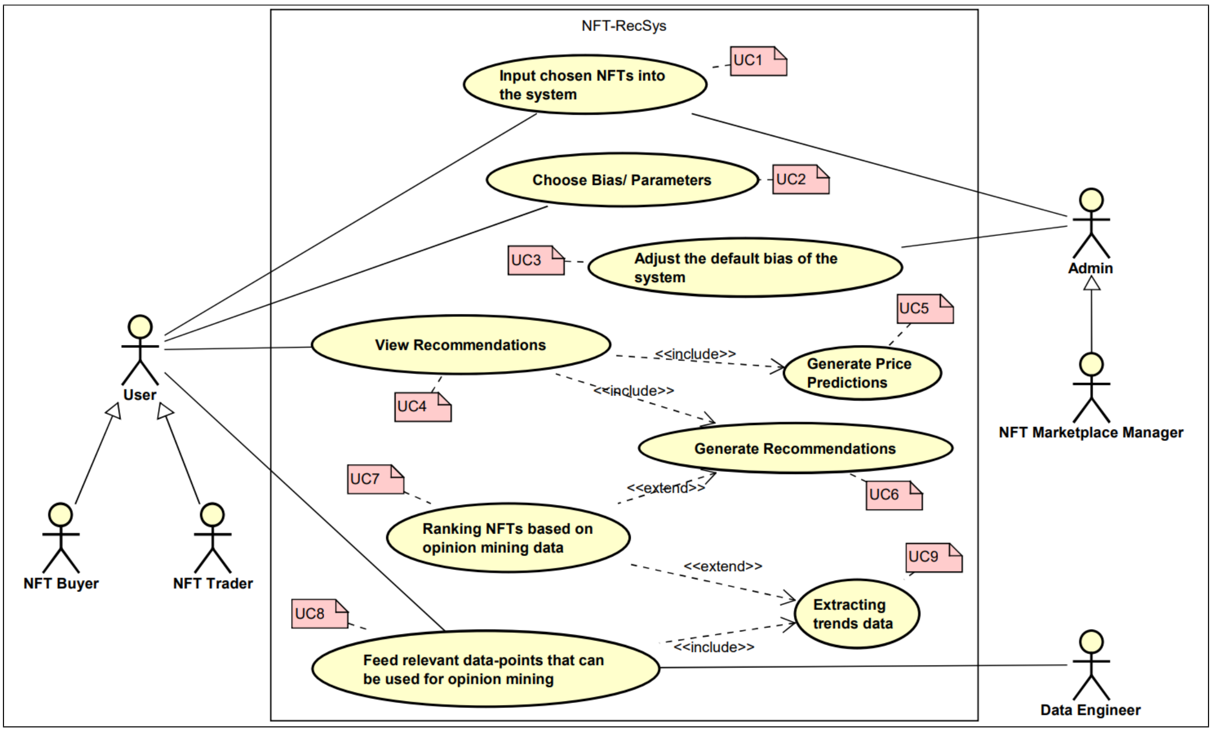


Figure 2.4: Use Case Diagram *(self-composed)*

2.9 Use Case Descriptions

Table 2.7: Use case description UC:04

Use Case	View Recommendations
Id	UC:04
Description	Display the most relevant NFT Recommendations based on the user's selection & available data in the system.
Primary Actor	User
Supporting Actors (if any)	none
Stakeholders and Interests (if any)	Admins, NFT Traders, NFT creator
Pre–Conditions	The NFT data and trends data have to have been pre-processed. The recommendations have to have been generated.
Post Conditions	Success end condition: The user is presented with recommended NFTs.

Trigger	A user wishes to find similar NFTTs to those that are currently being viewed or to explore possible interests based on past views.
Main Success Scenario	• User chooses the option to view recommendations. • System recognizes the user's preferred bias for recommendations. • System filters out and diversifies recommendations based on the user-bias and general bias that has been set in the system. • System displays the recommended NFTs.
Variations	A user can be presented with recommended NFTs based on past interests shown and views in a feed similar to a social network/ e-commerce store.

Table 2.8: Use case description UC:07

Use Case	Ranking NFTs based on Opinion mining data
Id	UC:07
Description	Rank NFTs for recommendations based on gathered social media trends data, opinion mining data & content in NFTs.
Primary Actor	none
Supporting Actors (if any)	Admins, Users
Stakeholders and Interests (if any)	NFT Collectors, NFT Traders, NFT creator
Pre–Conditions	New data-points have been added by an admin or a user and the trends have been extracted.
Post Conditions	Success end condition: Rank NFTs
Trigger	An admin or a user wishes to find NFTs that have content related to what's trending on the internet at the current moment in time.
Main Success Scenario	• System matches data of each NFT in the current data-set with extracted trends data. • System calculates a score for each NFT based on the matches & impact of the identified trends. • System re-ranks NFTs based on the calculated scores.

Variations	When recommendations are produced using other methods apart from trends, the data ranking scores generated here can be used to re-rank the recommendations when presenting to a user.

2.10 Requirements

2.10.1 Functional Requirements

The MoSCoW technique was used to determine the priority levels of system needs based on their importance.

Table 2.9: Levels of priority according to the "MoSCoW" technique.

Priority Level	**Description**
Must have (M)	This level's requirement is a prototype's core functional requirement, and it must be implemented.
Should have (S)	Important requirements aren't absolutely necessary for the expected prototype to work, but they do add a lot of value.
Could have (C)	Desirable requirements that are optional and aren't deemed essential critical to the project's scope.
Will not have (W)	The requirements that the system may not have and that are not considered a top priority at this time.

Table 2.10: Functional requirements

FR ID	**Requirement**	**Priority Level**	**Use Case**
FR1	Users must be able to add a chosen NFT to be considered as the reference point to generating recommendations.	M	UC1
FR2	Admins should be able to add a collection of NFT to be used as recommendations.	S	UC1
FR3	The system could be able to fetch relevant data of the NFT using an entered token Id.	C	UC1
FR4	Users must be able to set/ adjust the bias and parameters to be used by the Recommendations System using parametric selections prior to generating recommendations.	M	UC2

FR5	Admins should be able to adjust the default bias of the Recommendations System.	S	UC3
FR6	Users must be able to view recommendations with the click of a button.	M	UC4
FR7	The prototype could have an option to receive user feedback regarding the satisfaction level of the generated recommendations by the system.	C	UC4
FR8	The system could show the reasons for recommending each item to users.	C	UC4
FR9	The system should generate price predictions and consider the results for recommendations.	S	UC5
FR10	Opinion mining trends data must be used to generate NFT recommendations.	M	UC7
FR11	A user could be allowed to feed data-points such as interested public figures, websites to use as opinion mining data for recommendations.	C	UC8
FR12	Admins should be able to feed data-points such as interested public figures, websites to use as opinion mining data for recommendations.	S	UC8
FR13	User-input could be aggregated and used as a reinforcement learning bias for the Recommendations Model.	C	
FR14	The system will not act as a decentralized system.	W	

2.10.2 Non-functional Requirements

Table 2.11: Non-functional requirements

NFR ID	Requirement	Description	Priority Level
1	Performance	Although recommendations should be provided upon user-input; the recommendations matrix & opinion-mining data can be pre-processed and stored in-memory to be used. Real-time processing isn't essential.	Desirable
2	Quality of Output	The quality of the output should be of the highest possible level, utilizing all the available data.	Important

3	Security	The application should prevent any attackers from manipulating results and extracting user-inputs. Security could be assured by means of testing.	Desirable
4	Usability	Since the purpose of the system is to automate and make it easy for the user to explore NFTs, the usability of the system must be easy for users of all levels of expertise.	Important
5	Scalability	The prototype may open up for testing for many users. Considering the hype around NFTs and the interest in the project, the system may have to support many concurrent user-requests.	Desirable

2.11 Chapter Summary

In this chapter, a Rich Picture Diagram was drawn to illustrate how the system connects with the society to understand the stakeholders of the system. Saunder's Onion model was used to represent the stakeholders with the flow of influence of each stakeholder. Requirement gathering techniques were utilized to gather all the required data and opinions of possible stakeholders of the system. Lastly, the system's use cases, functional, and non-functional requirements were specified based on the insights derived from the requirement elicitation techniques.

CHAPTER 3: INITIAL DESIGN

3.1 Chapter Overview

This chapter consists of the design decisions made to come up with a suitable architecture for implementation, based on the gathered requirements. High-level design, low-level design, design diagrams, UI wireframes have been used to convey how the design goals are expected to be achieved while discussing the reasoning for chosen design decisions.

3.2 Design Goals

Table 3.1: Design Goals of the proposed system

Design Goal	**Description**
Performance	The recommendations matrix & opinion-mining data can be pre-processed and stored in-memory to be used for recommendations. Since ensembled models are expected to be utilized, concurrency would be ideal to get the output from multiple models at the same time. This could cut down the processing time by 4-5 times (based on the number of models that are required to provide recommendations for the given input).
Correctness	The correctness & quality of the output should be of the highest possible level, utilizing all the available data. By explaining why a user is getting the proposed recommendation will ensure that the user isn't mislead into wrong purchase decisions.
Usability	Since the purpose of the system is to automate and make it easy for the user to explore NFTs, the usability of the system must be easy for users of all levels of expertise.
Scalability	The system may have to support many concurrent user-requests in a production environment. The backend should be able to handle this. New data should be able to be added to the system with minimum effort.
Adaptability	Since the utilized Recommendation models may have to be altered based on the available data and user-requirements in the future, these models should be able to be easily swapped out for new models while ensuring that the system won't break in the process of upgrading, with minimum changes.

3.3 High-Level Design

3.3.1 Tiered Architecture

The system's architecture is depicted in the diagram below. The data, logic, and presentation layers are organized in a three-tier architecture.

The research contribution in this system lies in data preprocessing of the *data tier*, recommendations models and in the recommendations diversifier of the *logic tier*.

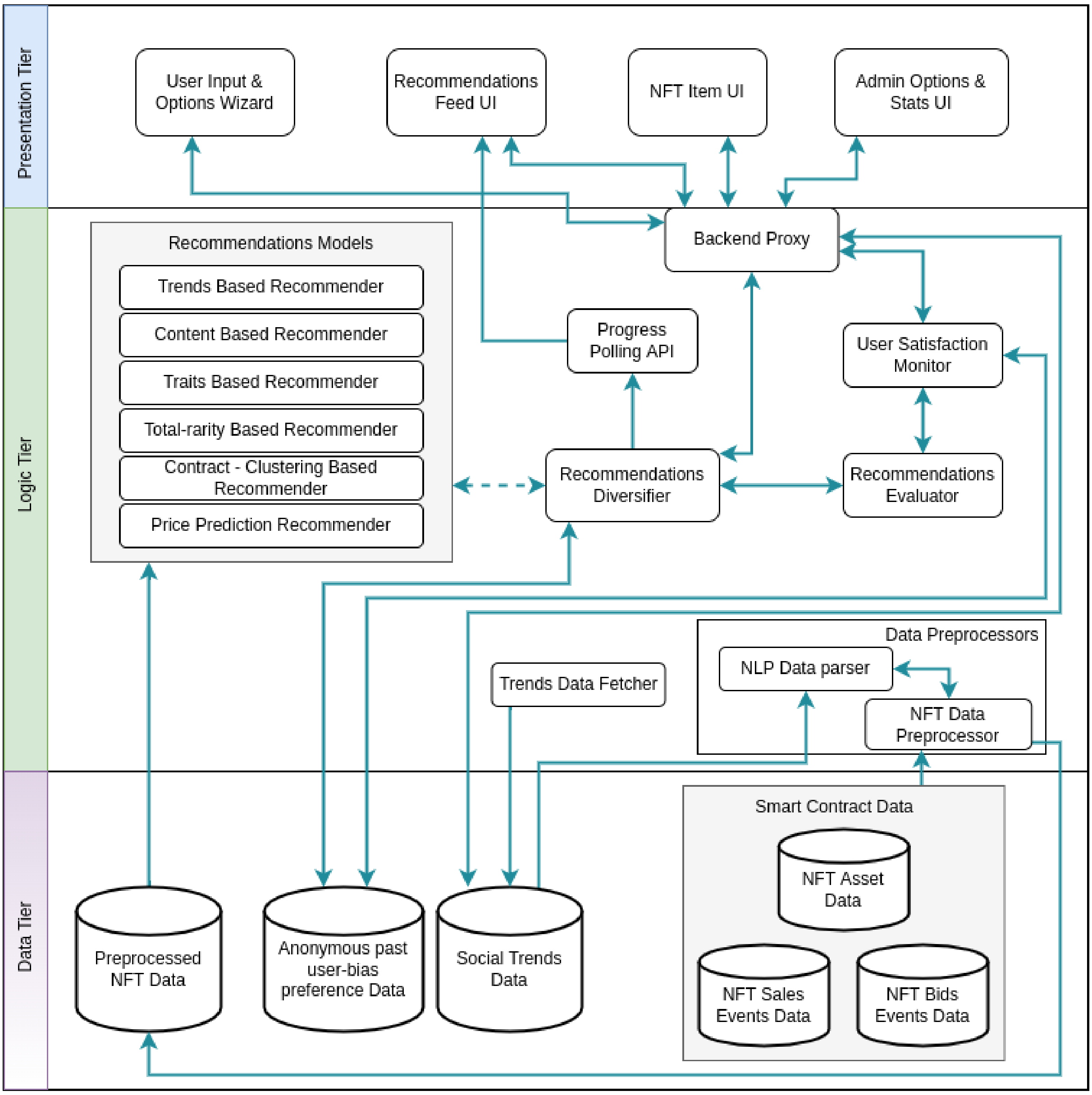


Figure 3.1: Three Tiered Architecture *(self-composed)*

While the entire architecture is represented in a modular approach for ease of understanding, several backend services are expected to work together in the fashion of a distributed microservices architecture when it comes to implementating the proposed architecture.

The reason for following a microservices architecture is to allow the system to scale while ensuring that points of failure can be easily recognized and taken care of seperately. The distributed nature of the system is expected to be seen in the connection between the numerous Recommendations Models and the Recommendations Diversifier. These combined together through output-pipelines, will act as an Ensebled Recommendations System. Although the system will be capable of distributing the load at this point, the expectation with the prototype is to run this in a single machine.

The purpose of each module that is represented in the above architecture are described below.

Data Tier

1. Smart Contract Data - Data that is retrieved from Blockchain Smart Contracts. For convenience purposes, the data is fetched from the OpenSea API. Contains all the available data of each NFT.
   (a) NFT Asset Data - All the content of each NFT.
   (b) NFT Sales Events Data - Past sales data from NFT trading.
   (c) NFT Bids Events Data - All the current bids of each NFT.
2. Social Trends Data - Data gathered from social trends sites (Twitter, news sites, etc.)
3. Anonymous past user-bias preference Data - Each user's preferred bias stored anonymously. This can be identified by a user's selection based on their requirement or based on the feedback received for each recommendation. This can be a temporary data-store that can be cleared once the user-session has ended.

Logic Tier

1. Data Preprocessors - The preprocessing code required to modify/ extract required data that is usable for recommendations from all the available data.
   (a) NLP Data parser - Responsible for extracting all the required data from what was collected through data mining techniques.
   (b) NFT Data Preprocessor - Used to modify and separate data that can be utilized from smart contracts and processed trends data.
2. Recommendations Models - The various models that are used to provide recommendations based on identified diverse data-points.
3. Recommendations Diversifier - The module that combines the recommendations produced by all the Recommendations Models, considering the bias.
4. User Satisfaction Monitor - The feedback received by user's will be filtered and updated through this module, to update the moving bias while preserving user-anonymity,

5. Recommendations Evaluator - The module that evaluates the user's satisfaction with the recommendations produced, to separately identify under-performing & high-performing models.
6. Progress Polling API - The web-polling API that will be used to update the progress of recommendations generation in the frontend.
7. Backend Proxy - The interface that exposes the backend services to the frontend.
8. Trends Data Fetcher - Fetch global trends data from social APIs or by scanning through news websites.

Presentation Tier (Client Tier)

1. User Input & Options Wizard - The UI that is presented to the user to enter the desired NFT(s) to be considered to recommendations as well as desired parameters and data-points (for advanced users).
2. Recommendations Feed UI - The UI that will show all the recommendations generated for a user. This will be similar to a home page on Youtube/ any other social network.
3. NFT Item UI - The UI that will show a chosen NFT with it's data and recommendations.
4. Admin Options & Stats UI - The UI that will be exposed to a system Admin, allowing him to view the stats such as the general bias of the system. This will have options to defined the data-sources to be used for trends based recommendations and to adjust the bias.

3.4 System Design

3.4.1 Choice of the Design Paradigm

Although the author was very tempted to use OOAD (Object Oriented Analysis and Design) to build the prototype due to the ease of extendability and further development of the system, the decision was made to use **SSADM (Structured Systems Analysis and Design Method)** based on the following factors.

- The project's core research component being inclined towards Data science. Therefore, it doesn't gain a noticeable benefit by using Object Oriented approaches.
- The programming languages that are expected to be used for implementation don't support OOP by nature.
- Ease of implementation of a MVP (Minimum Viable Product) for demonstrating the research application using the prototype.
- The time constraint of having to implement & document a research within the time span of 10 months.

3.4.2 Data Flow Diagram

The Level 1 Data Flow Diagram presented below provides a more extensive breakdown of the components of the Context Diagram that was presented in the SRS.

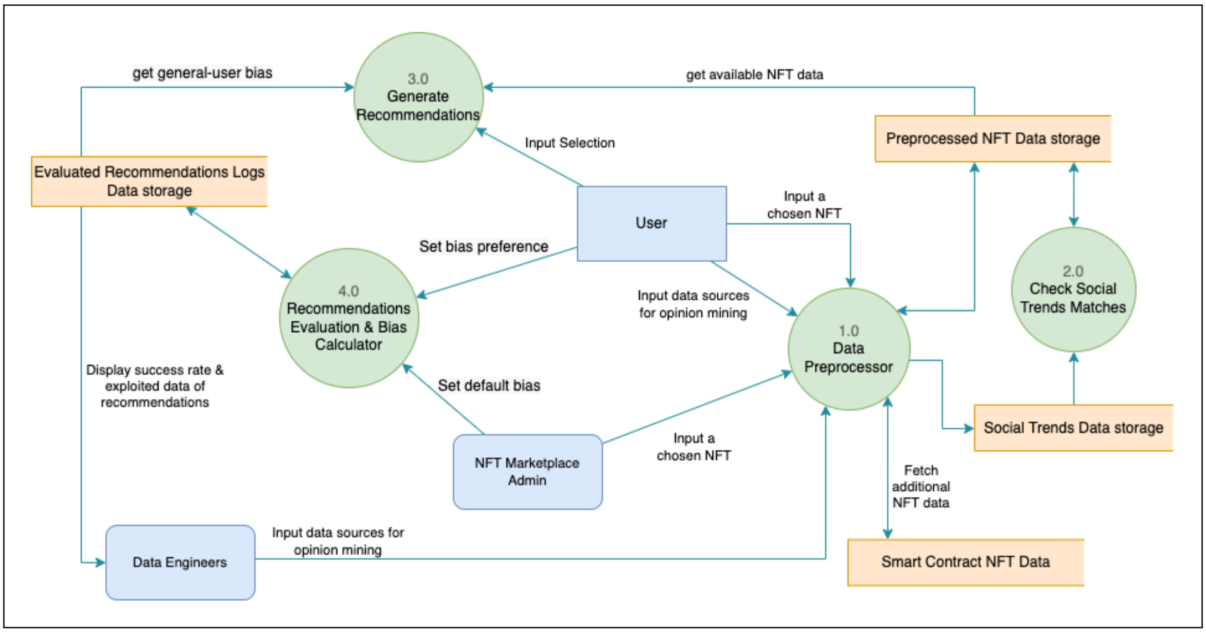


Figure 3.2: Data Flow Diagram - Level 1 *(self-composed)*

The Level 2 Data Flow Diagram presented below provides a more extensive breakdown of the components of the above Level 1 Data Flow Diagram.

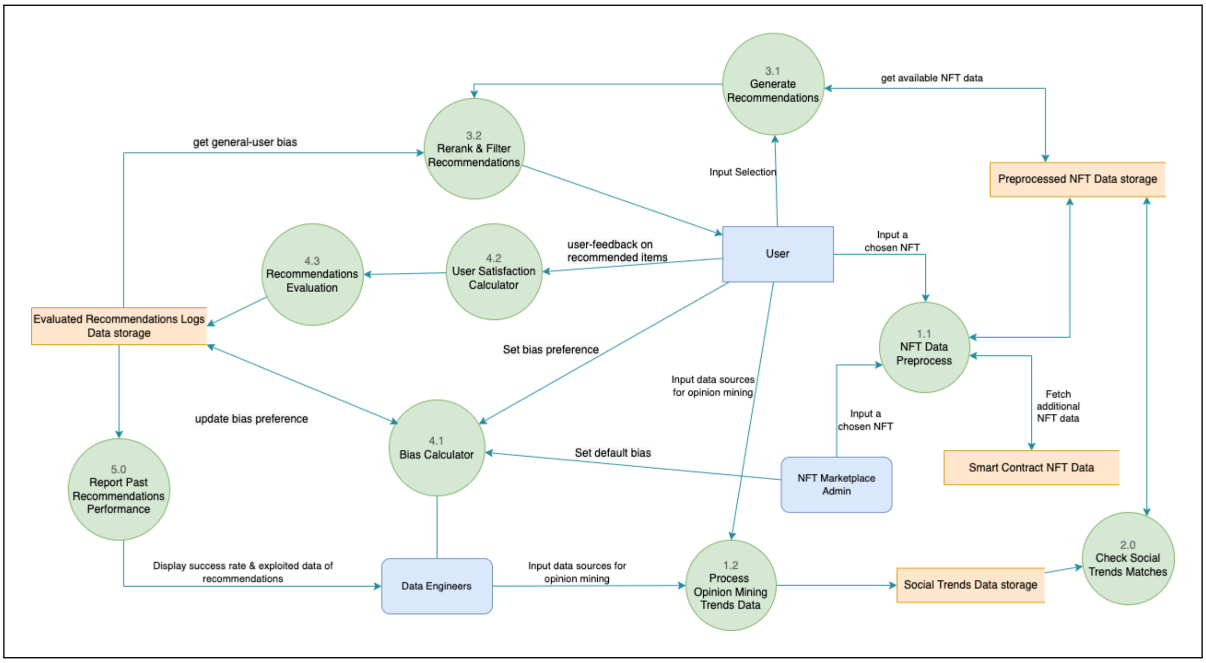


Figure 3.3: Data Flow Diagram - Level 2 *(self-composed)*

3.4.3 Algorithm Design

When studying available data in the system, it was identified that cross-collection NFTs cannot be recommended using the same concepts & data points followed for inter-collection matches. Therefore, multiple algorithms were considered to get a diverse set of recommendations.

Infusing trends matches into Recommendations

The equation composed below is designed to be used to calculate the total trends score for an item. The methods of utilization of this score for recommendations have been discussed following the breakdown of the equation.

$$T_{t_s,i} = \frac{\sum_{i_s=1}^{N_{i_s}} \left[\sum_{k_w=1}^{k_w} s_c \left(\frac{t_{vt,c}}{Med(T_{vt})} \right) \frac{mu}{(\mu+n_m)} \right]}{N_{i_s}} \tag{3.1}$$

Equation for social trend-match score for recommendations *(self-composed)*

$T_{t_s,i}$ - Total trends score for one item

N_{i_s} - Total number of information sources

i_s - Source of information

k_w - Number of keywords in the current item

s_c - Sentiment score surrounding chosen trend content

m - Match value, a Boolean used to check if the current evaluated content contains the chosen trend to be matched against.

u - User priority, used to check the current user's interest in the chosen trend. This is 1 by default

$t_{vt,c}$ - Tweet volume at this moment in time of the chosen content

$Med(T_{vt})$ - Median Tweet volume at this moment in time

μ - Constant, set to 0.1 to avoid division by 0 error for today's trends

n_m - Number of days between the current day & the day of the trend.

The following equation extracts the calculation of the impact score of the chosen trend (i_t), as described above. Twitter data has been taken as the example source here. The data source can be even an internet forum.

$$i_t = \frac{t_{vt,c}}{Med(T_{vt})} \tag{3.2}$$

Equation for the calculation of the impact score of a chosen trend *(self-composed)*

For trends that don't have a measurable volume, $t_{vt,c}$ can be taken as $(T_{vt}min - 1)$ to give it the lowest possible value, or as $Med(T_{vt})$ to omit the impact score all-together.

The algorithm, $T_{t_s,i}$ can be applied to inter-collection recommendations as well, if each NFT in the collection has unique names and descriptions. Using unique traits didn't seem to make sense for comparison with this algorithm, but it may be valid if it can be proved that the traits can be matched with trends data.

The Total trends score for one item calculated above can either be taken for recommendations as the top N items or as an absolute similarity match with other chosen items' trends scores.

The beauty of this equation is that it isn't necessarily required to be applied for only NFT recommendations. It can be used to enhance any content-based recommendations model. It can be seen as another way of infusing collaborative filtering, without the collection of user-specific data by the platform that integrates the presented Recommendations Architecture.

Recommendations based on Rarity

$$T_{r,t} = \sum_{t=1}^{Nt} \frac{1}{\left(\frac{c_t}{T_N}\right)} \tag{3.3}$$

Equation for the calculation of the total trait rarity score of an NFT *()*

$T_{r,t}$ - Total rarity of a trait

Nt - Total number of traits in the NFT

c_t - Trait count of the chosen trait (number of occurrences in the collection) T_N - Total supply of NFTs in the collection

The absolute difference between the total rarities are calculated when a NFT from a collection is chosen. The lowest scoring items are recommended to the user. This gives the NFTs that maybe as closely valuable as the initially chosen NFT. This allows to recommend NFTs that don't have unique content descriptions.

Furthermore, the traits are fed into a Content-based Recommendations Model to get NFTs with the most similar traits to be recommended.

Varying Bias for Recommendations Diversifier

Finally, all these recommendations produced by algorithmic models had to be presented to the user in a suitable manner. Instead of going with a weighted-bias which was recommended by the experts that were interviewed, it was decided to make this bias variable with time.

The reason for opting for this in-contrast to having a pre-trained weights & biases using a Neural Network architecture that Amazon successfully attempted with it's recent Autoencoder (Larry, 2019) DL model was to allow a more optimized output, without having to retrain the model. Another reason to opt for this method was because due to the lack of user-data to identify the most optimum weights or to train a DL model.

The calculation of this bias draws concepts from Reinforcement learning techniques.

$$B_{w,p} = \frac{\left[\sum_{i=0}^{n_g} \frac{b_{p,s}}{(\alpha+n_m)}\right]}{N_{n_g}} \tag{3.4}$$

Equation for the calculation of the recommendations bias in combining outputs in ensembled models *(self-composed)*

$B_{w,p}$ - Default Bias weighting for a chosen pipeline that recommendations are given from

$b_{p,s}$ - Successful bias selection for a chosen pipeline for the last n days

α - Constant, set to 0.001 to avoid division by 0 error for today's bias selections

n_m - Number of days away from the current day.

n_g - Grouped days (Eg: 1 day, 7 days, 1 month, 3 months, 6 months, 1 year)

N_{n_g} - Total number of grouped days considered

Applying Bias Push

When presenting recommendations, the author decided to allow a system admin to be capable of suggesting a push towards a preferred direction to allow the bias to be altered.

$$B_{c,p} = b_{l,p} + \left(B_{w,p} - b_{a,p}\right) \tag{3.5}$$

Equation for the calculation of the recommendations bias in combining outputs in ensembled models *(self-composed)*

$B_{c,p}$ - Current bias of a chosen recommendations pipeline

$b_{l,p}$ - Last applied user bias for the chosen recommendations pipeline. This can be 0 or null

$B_{w,p}$ - Default bias of a chosen recommendations pipeline

$b_{a,p}$ - Admin suggested bias of a chosen recommendations pipeline

The above bias will be applied only to user's who haven't chosen a preferred bias. It can be applied to users who have chosen the bias as well, but it is suggested to be applied after initially showing recommendations to the user using their requested bias.

3.4.4 UI Design

3.4.5 System Process Flow Chart

The algorithm's flow and decision structures are depicted in the flowchart below. It explains a significant proportion of the system since the expected implementation is primarily procedural.

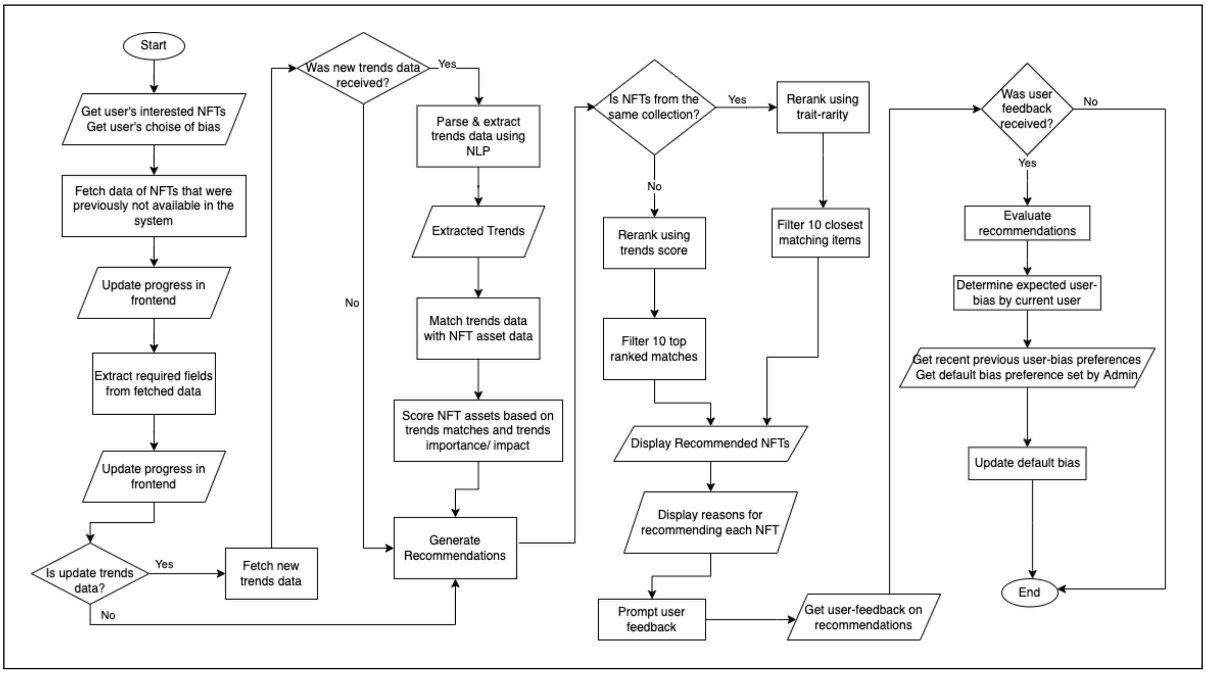


Figure 3.4: System Process Flow Chart*(self-composed)*

3.5 Chapter Summary

The design, architectural aspects and the flow of the project and novel author-designed algorithms were documented in this chapter followed by the expected UI wireframes to be implemented for the end-user's interaction with the system.

CHAPTER 4: INITIAL IMPLEMENTATION

4.1 Chapter Overview

This chapter explains the core implementation of the research prototype together with the technologies, languages & supporting tools used for development of the prototype, with reasoning to the choice of each selection.

4.2 Technology Selection

4.2.1 Technology Stack

The technologies that were used to implement the prototype at each layer are shown below.

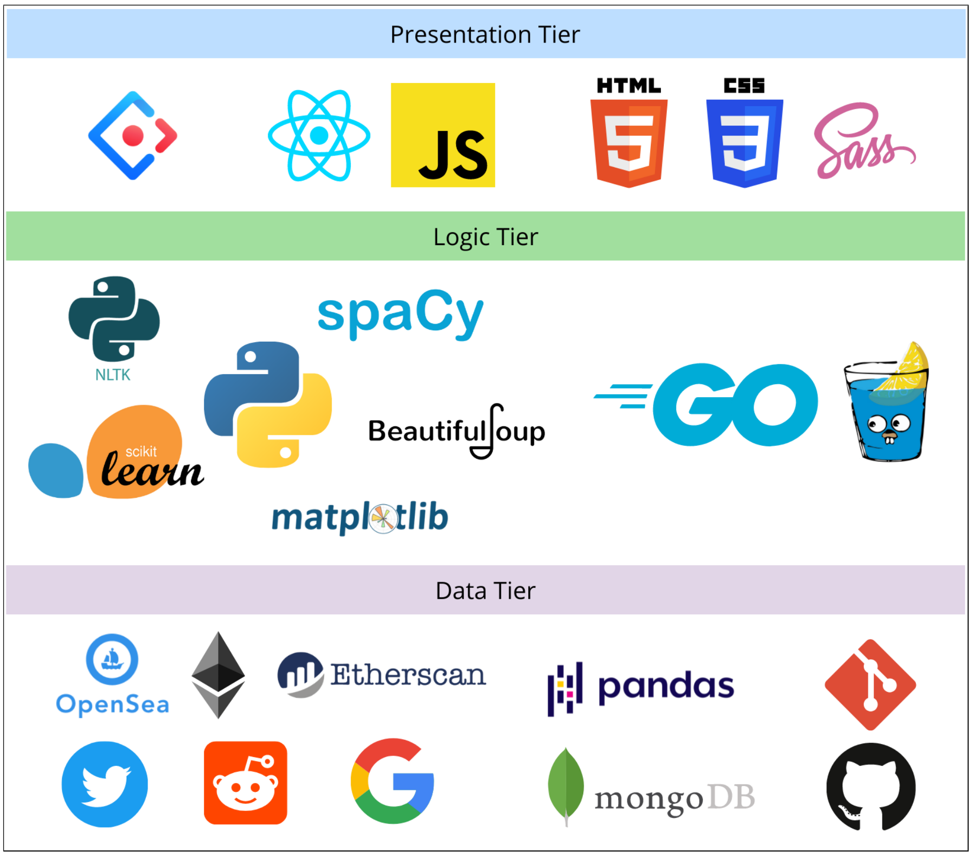


Figure 4.1: Technology Stack

Linux will be the default choice for development since of the ease of support for multiple development tools and performance benefits. MacOS/ Windows will be used for research documentation & study purposes.

The rest of the choices in the above tech-stack have been explained in the following sections.

4.2.2 Data Selection

Being a data science project at the core, it was important to choose the best possible sources of data to gather sufficient data for analysis & produce the best possible recommendations.

The data requirements identified were,

1. NFT asset data
2. Global trends data
3. NFT Smart Contract data
4. NFT events (sales) data
5. NFT bids data

Since the main technological research gap to be addressed was with the integration of global trends into content based recommendations, this was given a higher priority at first. These data requirements were sourced from the following sources and heavily pre-processed there after to create a usable dataset for data analysis.

- NFT asset, events, bids data - From the **OpenSea API**.
- Global trends data
  - Twitter data - From **Twitter developer API**.
  - Google Trends data - From Google Dataset Search & unofficial **Google Trends Python API (Pytrends)**.
- Ethereum Smart Contract data - From Etherscan & OpenSea

All the data-points that could be used for recommendations and explored with iterative development, as a research. This iterative process took a long time since the APIs were rate limited. The gathered pre-processed datasets will be made available for public use for future researches.

4.2.3 Selection of development framework

Table 4.1: Selection of development framework

Framework	**Justification for selection**
Gin Gonic	It's extremely convenient to build APIs using Gin with Golang. It also has an easily debuggable log output & claims smashing performance (up to 40 times faster!)

Ant Design	The world's second most popular React UI framework. Used in many industrial applications and has a wide range of components to match most UI requirements. Since it's tree-shaking compatible, it will build only the components that are used. This reduces build time of the frontend. The CSS is easily customizable as well.

Although this is a data science project, all data science models utilized were built from scratch without the use of libraries, since doing so allowed the author to tweak the models at will.

4.2.4 Programming language

Python is the language that will be used to create the ML models. Python is an all-purpose language that has been used in many projects involving data science. It has a vast collection of supporting libraries that eases many data science related tasks.

For the API proxy it was decided to use **Golang**, which is statically typed language that attempts to resemble the performance of C. Golang will allow the application to support concurrency and multi-threaded communications while being extremely lightweight and fast. This will be used to avoid any bottlenecks that could occur at this point in the system, while potentially bolstering performance.

For the frontend, **JavaScript** was decided to be used to show dynamic content and allow a highly interactible & inviting user experience.

4.2.5 Libraries Utilized

Table 4.2: Libraries Utilized with justification for choices

Library	**Justification for selection**
Pandas	Pandas dataframes allow a vast range of functionalities required for data analysis such as cleaning, transforming, filtering, sorting & manipulating of data
Scikit-learn	Used for vectorizing text and generate similarity matrices between items, for recommendations.
NLTK	Convenient to use for NLP data parsing, using the RAKE vectorizer.
SpaCy	Allows production-ready advanced NLP.
Beautiful Soup	Convenient to scrape data from the internet.
Matplotlib	Has almost any type of visualization method for data analysis.

React	A UI library that makes it easy to build interactive websites. Used as an alternative to using a framework since the vast array of capabilities and other integratable frameworks and libraries. It was important to develop an easily interactible frontend, since it will be the users' point of interaction with the system.

4.2.6 IDE's Utilized

Table 4.3: IDEs Utilized with justification for choices

IDE	**Justification for selection**
Google Colab	Convenience of trial & error of fetching data, building, testing ML models and ability to work across multiple devices with the cloud development environment.
VSCode	Extremely dynamic while being simple to use, yet powerful for front-end development with it's extensions & code snippets.
Golang	Convenient syntax highlighting & auto-completion for Golang development.
PyCharm	Well-equipped Python Integrated Development Environment (IDE) with a lot of capabilities.

4.2.7 Summary of Technology selection

Table 4.4: Summary of Technology selection

Component	**Tools**
Programming Languages	Python, Golang, JavaScript
Development Framework	Gin Gonic
UI Framework	Ant Design of React
Libraries	Pandas, Scikit-learn, NLTK, SpaCy, Beautiful Soup, Matplotlib, React
IDE – Research	Google Colab
IDE – Product	VSCode, Golang, Pycharm
Version Control	Git, GitHub
Application hosting	Netlify, AWS

4.3 Implementation of Core Functionalities

Since a Recommendations System's ultimate goal is to reduce the amount of information overload and provide the user with the best possible options, it was essential to build a dataset to suit the expected requirements. Just throwing in all the data fetched from APIs into a DL wouldn't give an expected successful recommendation. Therefore, the fetched data was heavily preprocessed.

NFT Data Mining

Continuously being able to add new NFTs or even adding an initial set of NFTs should be possible in the system for users' convenience. When doing so, we need to make sure that relevant information is extracted.

Figure 4.2: Implementation code segment: NFT data mining & preprocessing

The data extraction is done to extract information required for recommendations, to view details of items & to save information for recommendation algorithms/ predictions that are potentially possible in the future.

NLP Preprocessing, Vectorizing & Recommendations

```
# instantiating and generating the count matrix
count = CountVectorizer()   # used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text
count_matrix = count.fit_transform(df['All_key_words_str'])
```

Figure 4.3: Implementation code segment: Content Vectorizer

A Count Vectorizer was used from the *scikit learn* library to vectorize all words, to be used for similarity matching. The reason for choosing the Count Vectorizer over a Tf-Idf Vectorizer was because Tf-Idf will give lower scores to more common words found in the dataset. Since our intent is to identify all the possible matches and primarily rank the content based results using global trends, it made more sense to go with a Count Vectorizer.

A Cosine Similarity Matrix is then generated from the *scikit learn* library to identify all the matching words contained across all NFTs content. This generates the recommendation ahead of time.

```
# generating the cosine similarity matrix
cosine_sim = cosine_similarity(count_matrix, count_matrix)
```

Figure 4.4: Implementation code segment: Generating the Cosine Similarity Matrix

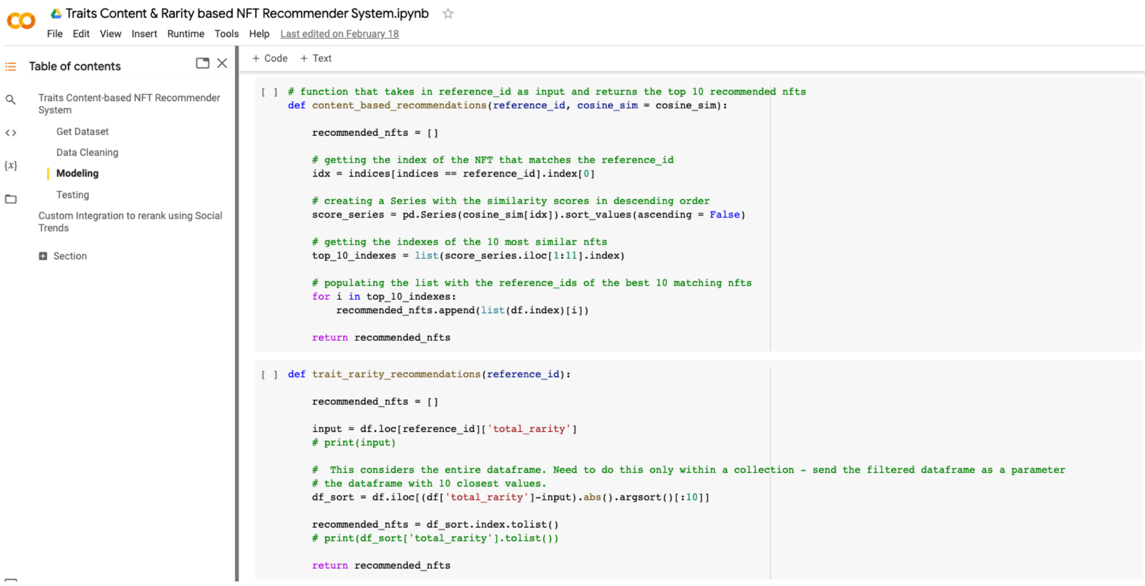


Figure 4.5: Implementation code segment: Produce Trait Rarity Based Recommendations

The above recommendation generation algorithms were created to cater towards matching NFTs within a collection, since most of the major NFT-collections have comparatively more unique data in traits compared to descriptions. Trait rarity similarity was identified to be the best way to identify total uniqueness which represents the value of each NFT. Although the calculation of total rarity was explored by *rarity tools* during the course of the research, recommending similar total rarities is a novel implementation in the application domain.

Trends Extraction, Preprocessing & Recommendations

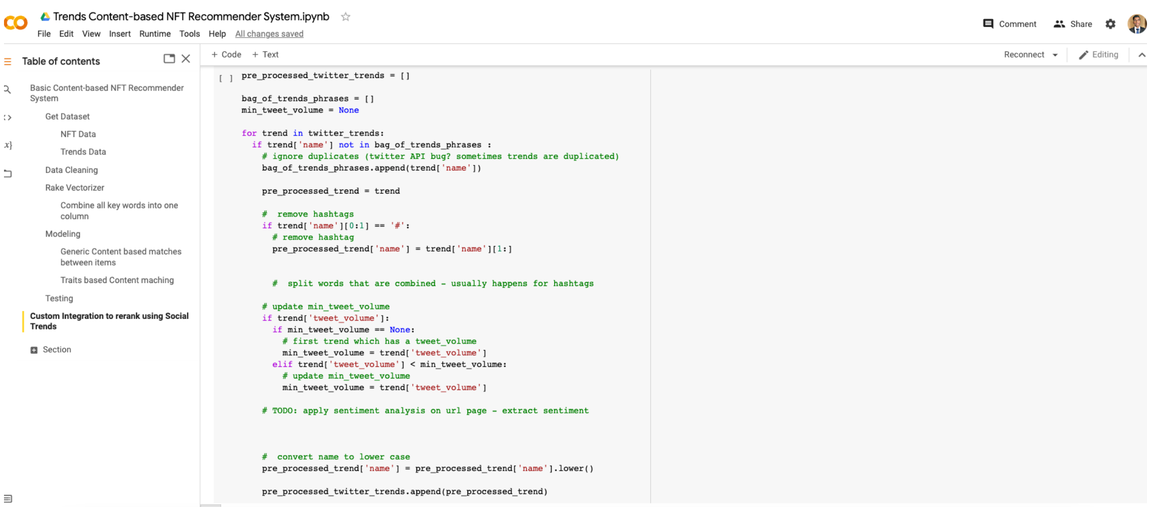


Figure 4.6: Implementation code segment: Preprocess Trends Data

The above code segment preprcoesses trends that are fetched from the live Twitter API.

Figure 4.7: Implementation code segment: Calculating Trends Score

The above code segment assigns a tweet volume for trends with no volume & calculates the median Tweet volume which used to calculate the impact score of each trend.

The below code segment is used to calculate the trends score for each NFT and finally make trends-based recommendations.

Figure 4.8: Implementation code segment: Calculating Trends Score

4.4 Self-Reflection

The current prototype implementation covers the core research component focused in the research, but there're point of improvements that the author would like to achieve before the end of the final prototype. The use of multiple data sources for trends is something that can be added as a plugin, to increase the trends-score and find more matches for trends based recommendations. Furthermore, the utilization of item-to-item collaborative filtering and price prediction would be a great addition, to complete & present the recommendations ecosystem that is possible to be created with the suggested design architecture.

Prior to adding more models to the system, the author's primary goal in implementation over the next few weeks would be to implement the front-end, API proxy and database connection to present a completed, user-friendly minimum viable product.

Considering the tight deadline and time-constraint to achieve a completed core research component, the level of completion of the prototype together with the research documentation is extremely satisfactory.

Embarrassing concepts of Decentralized Systems & paving the path to Web 3.0, the research contribution brought forward in this project is expected to open up even more possibilities with time.

4.5 Video Demo

The link to the demo video presenting the current implementation progress can be found here: `https://www.youtube.com`

4.6 Chapter Summary

The chapter comprised of the technologies, languages & supporting tools utilized to implement the prototype developed as part of the research. Discussions accompany the code snippets and algorithms produced as part of core functionality. Finally, the author's self-reflection of the developed prototype was presented.

REFERENCES

Ayushi, Smriti and Badri Prasad (Nov. 8, 2018). "Cross-Domain Recommendation Model based on Hybrid Approach". In: *International Journal of Modern Education and Computer Science* 10.11, pp. 36–42. ISSN: 20750161, 2075017X. DOI: 10.5815/ijmecs.2018.11.05. URL: http://www.mecs-press.org/ijmecs/ijmecs-v10-n11/v10n11-5.html (visited on 07/12/2021).

Chen, Rung-Ching and Hendry (June 2019). "User Rating Classification via Deep Belief Network Learning and Sentiment Analysis". In: *IEEE Transactions on Computational Social Systems* 6.3. Conference Name: IEEE Transactions on Computational Social Systems, pp. 535–546. ISSN: 2329-924X. DOI: 10.1109/TCSS.2019.2915543. (Visited on 05/25/2021).

Cheng, Li Chen and Ming-Chan Lin (Oct. 2020). "A hybrid recommender system for the mining of consumer preferences from their reviews". In: *Journal of Information Science* 46.5, pp. 664–682. ISSN: 0165-5515, 1741-6485. DOI: 10.1177/0165551519849510. URL: http://journals.sagepub.com/doi/10.1177/0165551519849510 (visited on 07/16/2021).

Chevet, Sylve (2018). "Blockchain Technology and Non-Fungible Tokens: Reshaping Value Chains in Creative Industries". In: *SSRN Electronic Journal*. ISSN: 1556-5068. DOI: 10.2139/ssrn.3212662. URL: https://www.ssrn.com/abstract=3212662 (visited on 04/18/2021).

Choi, Minjin et al. (Mar. 8, 2021). "Local Collaborative Autoencoders". In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. WSDM '21. New York, NY, USA: Association for Computing Machinery, pp. 734–742. ISBN: 978-1-4503-8297-7. DOI: 10.1145/3437963.3441808. URL: https://doi.org/10.1145/3437963.3441808 (visited on 05/18/2021).

Clark, Mitchell (Mar. 3, 2021). *People are spending millions on NFTs. What? Why?* The Verge. URL: https://www.theverge.com/22310188/nft-explainer-what-is-blockchain-crypto-art-faq (visited on 07/19/2021).

Conti, Robyn (Apr. 29, 2021). *What You Need To Know About Non-Fungible Tokens (NFTs)*. Forbes Advisor. Section: Investing. URL: https://www.forbes.com/advisor/investing/nft-non-fungible-token/ (visited on 07/19/2021).

Decentralized applications (dapps) (Sept. 1, 2021). ethereum.org. URL: https://ethereum.org (visited on 09/01/2021).

dfinzer (July 20, 2021). *Announcing our $100M raise, led by a16z*. OpenSea Blog. Section: Company Announcements. URL: https://opensea.io/blog/announcements/announcing-our-100m-raise-led-by-a16z/ (visited on 07/23/2021).

Dowling, Michael (Apr. 29, 2021a). "Fertile LAND: Pricing non-fungible tokens". In: *Finance Research Letters*, p. 102096. ISSN: 1544-6123. DOI: 10.1016/j.frl.2021.102096. URL: https://www.sciencedirect.com/science/article/pii/S154461232100177X (visited on 07/17/2021).

— (Apr. 29, 2021b). *Is non-fungible token pricing driven by cryptocurrencies? | Elsevier Enhanced Reader*. DOI: 10.1016/j.frl.2021.102097. URL: https://www.sciencedirect.com/science/article/pii/S1544612321001781?via%3Dihub (visited on 06/23/2021).

EIP-2309 (2021). *EIP-2309: ERC-721 Consecutive Transfer Extension*. Ethereum Improvement Proposals. URL: https://eips.ethereum.org/EIPS/eip-2309 (visited on 07/20/2021).

ERC (2021). Ethereum Improvement Proposals. URL: https://eips.ethereum.org/erc (visited on 08/24/2021).

ERC-20 Token Standard (2021). ethereum.org. URL: https://ethereum.org (visited on 08/31/2021).

ERC-721 Non-Fungible Token Standard (2021). ethereum.org. URL: https://ethereum.org (visited on 07/19/2021).

Fairfield, Joshua (Apr. 6, 2021). *Tokenized: The Law of Non-Fungible Tokens and Unique Digital Property*. SSRN Scholarly Paper ID 3821102. Rochester, NY: Social Science Research Network. URL: https://papers.ssrn.com/abstract=3821102 (visited on 04/18/2021).

Frankenfield, Jake (2021). *Decentralized Applications – dApps*. Investopedia. URL: https://www.investopedia.com/terms/d/decentralized-applications-dapps.asp (visited on 09/01/2021).

Hackett, Robert (Mar. 18, 2021). *This crypto marketplace just raised $23 million to be the 'Amazon of NFTs'*. Fortune. URL: https://fortune.com/2021/03/18/nft-art-crypto-marketplace-opensea-amazon/ (visited on 07/15/2021).

Introduction to smart contracts (2021). ethereum.org. URL: https://ethereum.org (visited on 08/31/2021).

Larry, Hardesty (Nov. 22, 2019). *The history of Amazon's recommendation algorithm*. Amazon Science. Section: Latest news. URL: https://www.amazon.science/the-history-of-amazons-recommendation-algorithm (visited on 05/25/2021).

Matney, Lucas (July 20, 2021). *NFT market OpenSea hits $1.5 billion valuation*. TechCrunch. URL: https://social.techcrunch.com/2021/07/20/nft-market-opensea-hits-1-5-billion-valuation/ (visited on 07/23/2021).

Naumov, Maxim et al. (May 31, 2019). "Deep Learning Recommendation Model for Personalization and Recommendation Systems". In: *arXiv:1906.00091 [cs]*. arXiv: 1906.00091. URL: http://arxiv.org/abs/1906.00091 (visited on 04/29/2021).

Off the chain (July 5, 2021). *Off the chain: NFT market surges to $2.5B so far this year*. Aljazeera. URL: https://www.aljazeera.com/economy/2021/7/5/off-the-chain-nft-market-surges-to-2-5b-so-far-in-2021 (visited on 07/15/2021).

Recommendations (2021). *Recommendations: What and Why? | Recommendation Systems*. URL: https://developers.google.com/machine-learning/recommendation/overview (visited on 08/24/2021).

Theorem, Takens (Jan. 30, 2020). *What are you missing? Using basic machine learning to predict and recommend NFTs with OpenSea data - OpenSea Blog*. URL: https://opensea.io/blog/analysis/predict-and-recommend-nfts/ (visited on 08/27/2021).

Vanderbilt, Tom (2021). "The Science Behind the Netflix Algorithms That Decide What You'll Watch Next". In: *Wired* (). Section: tags. ISSN: 1059-1028. URL: https://www.wired.com/2013/08/qq-netflix-algorithm/ (visited on 08/27/2021).

Wood, Dr Gavin (2014). “ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER”. In: p. 39. (Visited on 07/15/2021).

APPENDIX A - CONCEPT MAP

Figure 9: Concept Map *(self-composed)*