Jack and Panya

Knowledge Discovery From Data

04 September 2022

Machining Learning using an ID3 Decision Tree from a dataset of Dapp usage

Introduction and Background

For this assignment we decided to explore user trends on the Aptos blockchain and apply machine learning techniques we have gone over in the class to classify and predict users future actions. By grouping users into categories and doing analysis on their 'dapp' usage and making predictions about their future actions, dapps they might also be users of and their account balances. Before diving into the methods we applied and the results we came to, first we must give a background on the Aptos blockchain, including why we chose this chain to analyze and how this information might be valuable to builders(people producing blockchain apps), founders of the network, and users of the network.

Origins of the Aptos network and background

The Aptos network is a relatively new blockchain launched this year by the former members of Facebooks' Libra/Diem core team. Since its launch this October the platform has already secured its place on the top 20 blockchains by market cap and has over 100k users, becoming one of the fastest growing blockchains to date. This blockchain is highly experimental and in no way is this report attempting to endorse or promote this network, instead the network being in its infancy makes it an ideal candidate to explore user trends.

Exploring User Dapp Activities - Gathering the Data

In order to collect the user transaction info to catalog user usage of these applications we first needed a set of user wallet addresses. We have gathered a group of 30 thousand wallet addresses from a collection of protocols that published ‘whitelists’, users who have used the app in its testing phase or pre-official launch. We used these logs from apps such as LiquidSwap and Animeswap, both of these lists contained more than 10 thousand users each. Additionally we need the addresses where the applications are deployed, a semi-updated list is kept by seam.money, a site maintained by Jack, which allows users to inspect the modules(or functional capabilities) and resources of any address on the network. With the addresses of users and the locations we are monitoring for interactions, the next step is to collect the transactions of these users. To get the transactions of a user, we send requests containing the users address to an RPC node connected to the network. An RPC node acts as a gateway to the ledger of transactions verified by the network of nodes. After making individual requests to this node for each user for their 25 most recent transactions, we can begin parsing this info to determine what applications they are using. Every transaction consists of a sender, a payload, and a list of state changes, for our use case we just care about the sender and the payload. The payload contains the address of the function that the user is calling, we match these function addresses to their corresponding dapp and tally the counts for each user. With the now interpretable data, we are able to start creating our model to pull out interesting insights into user history in order to draw conclusions about the usage of these dapps.

Building and Testing the Model

After doing exploratory data analysis on small groups of users we began to apply a variety of different techniques to extract trends represented by these users' transactions. Starting off we tried using KNN to attempt to cluster the users based on their dapp preferences. Using a variety of values for K we got underwhelming results that didn't provide valuable insights into how we can break up these users into distinct categories. In attempting to classify these users by DeFi or NFT(based on the categories of the dapps they used) we observed that most of the users had little preference for one category or another and seemed to use both with similar frequency. However, from this we did get a hint that users of certain DeFi apps tended to use particular NFT apps with higher frequencies than the users of other DeFi apps. From here we decided to apply Naive Bayes to break down the likelihood of users of app 'B' being a user of app 'A'. By applying this technique we were able to get a better grasp on the distribution of users among these apps. With this technique we were able to successfully estimate additional dapps a user uses with 75% accuracy based on their least interacted with dapp. For verifying our model we are referencing transactions that have already occurred, in a real application this would give us insights into what new dapps a user may use next. This is not remarkable, given that the average user uses three dapps and we were monitoring for the use of 15 dapps, but we believe this approach offered more reputable and useful insights. When testing this technique, we are simply noting the use of one app, and finding the app which they are most likely to also be a user of. We then used these probabilities and published user counts of these dapps to estimate total users of each individual app. While we do not have the real user counts for all of these apps, by using the Liquidswap user count we were able to accurately estimate the user counts on Animeswap and Hippo Labs within 5% of their true values.

ID3 Decision Tree Implementation

Additionally we implemented an ID3 Decision tree to attempt to classify whether the user is a 'Big-shark' or a High roller. ID3 stand for iterative dichotomizer 3 and is named that way because the aglorthm iteratively deivides features into two or more groups at each step. It was implemented using a top-down approach which just means it obtains its information from the data first from the top node and builds its way down to the last leaf. Our ID3 decision tree algorithm uses entropy and information gain calculations. Entropy can be defined as the measure of disorder within a dataset; from my own interpretations, it means the randomness of a system. There is a correlation between high entropy, high order and high randomness. Information Gain is the calculations of the reduction in entropy and measures how well a feature classifies a targeted class. The highest information gain would be selected as the best one. Our algorithm determine which becomes the root node and then the corresponding nodes depending on which decision has the most information gain.

Entropy(S) = - ∑ p_i * $\log_2(p_i)$; i = 1 to n
n is the total number of attributes in the Predictors column

p_i is the **probability of class 'i'** or the ratio of "*number of rows with class i in the target column"* to the "*total number of rows"* in the dataset.

Example: Entropy(**Outlook) —> - ∑ Sunny * $\log_2$(Sunny) → -∑ ⅗* $\log_2$(⅗) + ...**

Information gain: **MAX(For every S: Entropy(Parent) - ∑Entropy(S))**

parent is the probability or ratio of how many **Big-Sharks** are in the dataset.

s is the **probability** of the **column attributes where big-shark is yes**.

Our Algorithm's Pseudocode:

ID3_Alogirthm : Matrix M

Input: M matrix

Output: N number

```
1. For every attribute in the matrix:
2.     Partial = get the partial matrix given only the attribute
3.     Total Entropies = [ ]
4.     For every name in the attribute:
5.         Parent_entropy = Entropy(P)
6.         Children_entropy = Entropy(S)
7.         Information_gain = Parent_entropy - Children_entropy
8.         Add information gain to total entropies
9.     If MAX(Total Entropies) == 0:
10.            Return 0
11.        ID3_Algorithm (Partial)
12.        Total Entropies = [ ]
```

Conclusion and relevancy of results

The ID3 Decision trees provided us with our most insightful results allowing easy classification of a users status based on minimal knowledge of their dapp usage. Additionally, the side-effect of being able to estimate user quantities moderately accurately was a pleasant unintended consequence of the probability distribution that resulted from the naive bayes implementation. This is unique because counting the users for all apps otherwise requires analyzing every transaction on the chain. Though our data didn't give us clear results for KNN clusters, it forced

us to be agile with our approach and try a variety of techniques which led us to some interesting discoveries. These approaches could be further explored and refined to offer valuable insights to dapp developers to help them more efficiently capture their target audiences and more clearly understand their userbases.

References

Seam Pass Whitelist:

https://docs.google.com/spreadsheets/d/1lZTHj5cOhOcoE4_G17cEMPzAKCGDiIu3ICCNB9p5yG8/edit#gid=0

Liquidswap whitelist:

https://docs.google.com/spreadsheets/d/e/2PACX-1vThwA1l64lib_QNppJFZNSssFotISGmmxN8oBgZyZ6Jmkx5tCEZoBsAaIUMMg44ZakiB6FraW-YyXkN/pubhtml

AnimeSwap whitelist:

https://docs.animeswap.org/docs/tutorial/Tokenomics