Dota 2 Hero Item Inventory Association

Seth Pavlicek (129852861) and Ziyad Kassar (825354160)

***Abstract*—In the multiplayer video game Dota 2 players' inventory are very complicated and can depend on many different factors. A player's inventory consists of 6 items and the items the player chooses heavily depends on the hero they play for that match. Unsupervised machine learning techniques were used in order to look for patterns in players' inventory.**

I. Introduction

Defense of the Ancients 2 (Dota 2) is a multiplayer video game where each match consists of 10 players divided into 2 teams, with 5 players per team. Each player selects and plays 1 unique hero, ensuring no 2 players can use the same hero. There are approximately 120 heroes to choose from. Additionally, there are over 200 items available in the game. Each player's inventory accommodates a maximum of 6 active items that provide gameplay benefits. Although players have 3 additional inventory slots, items placed in these slots do not provide any active or passive benefits during gameplay.

Items can be roughly grouped into a few groups. There are offensive, defensive, support, passive, and active items. These groups are not mutually exclusive. Some items provide stats which can boost a player's offense and defense. Support items can provide temporary bonuses. Passive items provide bonuses while they are equipped. Active items provide abilities that the player must activate.

Some items will be acquired by every hero no matter their role such as boots which provide movement speed. Other items such as daedalus are only used by certain heroes that are primarily focused on attack damage.

More specifically, the stats that items can provide are:

- Strength
- Dexterity
- Intelligence
- Damage
- Armor
- Spell Damage
- Maximum Health
- Lifesteal
- Spell Lifesteal
- Health Regeneration
- Attack Speed
- Cast Range
- Movement Speed
- Cooldown Reduction
- Maximum Mana
- Mana Regeneration

The OpenDota API was utilized to query approximately 1,000 matches, collecting data on player inventories at the end of each match. Each match generated ten rows of data, corresponding to the ten players involved.

The Open Dota API provides detailed information for any match. A single output from a match is formatted in a JSON file of over 15,000 lines of text due to all the metadata related to the game.

The important information that was parsed from the JSON includes:

- Hero IDs
- Item IDs

Association rule mining and clustering techniques were applied to this dataset to identify patterns and insights related to inventory choices and gameplay strategies. The analysis focused on uncovering relationships between item selections and their potential impact on match outcomes.

Fig. 1: WindRanger Hero

II. Dataset Description

The dataset consists of 13,851 rows and 7 columns, representing player inventory data. The

Fig. 2: Cheap Items

columns are as follows:

- `Hero_id` (integer): Identifies the hero used by the player.
- `Item_0` to `Item_5` (integers): Specifies the six items equipped by the player at the end of the match.

Additionally, the dataset was transformed using one-hot encoding, resulting in 303 columns. In the one-hot encoded data, values are binary (1 or 0), indicating the presence or absence of a specific item in a player's inventory. This encoding facilitates the application of machine learning algorithms for pattern recognition and analysis.

III. Models Used

Association rule mining and clustering techniques, specifically K-Means and DBSCAN, were employed to analyze the dataset. These models were selected due to the absence of labeled data, making supervised learning methods unsuitable. The high dimensionality of the dataset, resulting from the large number of items and heroes within the game, further necessitated the use of unsupervised learning techniques. These approaches enable the extraction of meaningful patterns and groupings from the data without predefined labels.

Fig. 3: Expensive Items

	item_98	item_96	item_94	item_931	item_92	item_911	item_908	item_907	item_90	item_9	...	hero_107	hero_106	hero_105	hero_104	hero_103	hero_102	hero_101	hero_100	hero_10	hero_1
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

Fig. 4: OneHotEncoded Dataframe

Evaluation metrics were used to assess the effectiveness of the applied models. Confidence and support were employed to measure the strength of the rules generated during association rule mining, providing insights into the reliability and frequency of item combinations. For clustering models, Sum of Squared Errors (SSE) was utilized to evaluate the quality of the clusters formed by K-Means and DBSCAN, indicating the compactness and separation of clusters within the high-dimensional dataset. These metrics ensure a robust evaluation of the patterns and groupings identified in the analysis.

IV. Results

A. Association Rules

The `mlxtend` library was used to generate association rules from the dataset, which required a one-hot encoded DataFrame. This transformation expanded the dataset from 7 columns to over 300 columns. Common items, such as most boot variants and Black King Bar (BKB), as well as items exclusive to in-game events, were excluded to focus on more distinctive patterns. An interesting rule identified was the frequent combination of `bloodstone`, `travel_boots`, and `bottle` with `hero_52`. Other heroes and their typical item sets did not appear prominently, likely due to the variability in itemization strategies based on gameplay situations. The observed rule highlights that, regardless of the player, `hero_52` consistently relies on the same set of items. Other similar patterns likely did not arise because players generally pick items to counter players on the opposite team. It could also be likely that more expensive items are less common and do not appear in the dataset frequently enough to make it above the minimum support threshold.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
0	(sheepstick)	(travel_boots)	0.022166	0.070975	0.007220	0.325733	4.589421
1	(travel_boots)	(sheepstick)	0.070975	0.022166	0.007220	0.101729	4.589421
2	(sheepstick)	(ultimate_scepter)	0.022166	0.150830	0.005271	0.237785	1.576507
3	(ultimate_scepter)	(sheepstick)	0.150830	0.022166	0.005271	0.034945	1.576507
4	(ogre_axe)	(mithril_hammer)	0.049386	0.018484	0.008014	0.162281	8.779640
5	(mithril_hammer)	(ogre_axe)	0.018484	0.049386	0.008014	0.433594	8.779640

Fig. 5: Top 6 Association Rules

B. Clustering

Clustering techniques, including K-Means and DBSCAN, were applied to analyze the dataset. Experiments with different values of k in K-Means generally resulted in poor clustering performance, as indicated by high within-cluster variance. DBSCAN provided better clustering results, although the number of clusters varied depending on the values of the epsilon and minimum samples parameters.

Figures 6 and 7 display the DBSCAN algorithm results in a different number of clusters as the minimum number of samples changed. It is likely that the clusters represent different patterns. For example the 169 clusters may represent per hero strategies because the number of clusters is close to the number of heroes. It should also be noted that some heroes can have different play styles that result in different inventories. The 7 clusters likely represent the different roles that are within the game.

```
dbscan = DBSCAN(eps=0.5, min_samples=10)
dbscan_labels = dbscan.fit_predict(scaled_data)
data['DBSCAN_Labels'] = dbscan_labels
[15] ✓ 1.2s

print("Number of clusters:", len(set(dbscan_labels)) - (1 if -1 in dbscan_labels else 0))
print("Number of noise points:", list(dbscan_labels).count(-1))
[16] ✓ 0.0s

Number of clusters: 7
Number of noise points: 13772
```

Fig. 6: Min 10 Clusters Results

```
dbscan = DBSCAN(eps=0.5, min_samples=4)
dbscan_labels = dbscan.fit_predict(scaled_data)
data['DBSCAN_Labels'] = dbscan_labels
[19] ✓ 1.2s

print("Number of clusters:", len(set(dbscan_labels)) - (1 if -1 in dbscan_labels else 0))
print("Number of noise points:", list(dbscan_labels).count(-1))
[20] ✓ 0.0s

Number of clusters: 169
Number of noise points: 13001
```

Fig. 7: Min 4 Clusters Results

To analyze the clusters, Principal Component Analysis (PCA) was used to reduce the data to two dimensions for better visualization and interpretability.

The K-Means algorithm was first applied with varying values for the number of clusters (k). This method generated distinct groupings, as seen in the visualizations (Figure 8). However, some overlap between clusters was observed, suggesting that while certain heroes and their inventories could be grouped based on patterns, other combinations were more ambiguous, leading to less distinct boundaries. This overlap reflects the dynamic and situational nature of item choices in Dota 2, where players adapt their inventories to match the specific conditions of each game.

Next, the DBSCAN algorithm, a density-based clustering method, was applied to the same PCA-reduced data. The DBSCAN results, shown in Figure 5, revealed fewer distinct clusters, with a significant portion of the data labeled as noise.

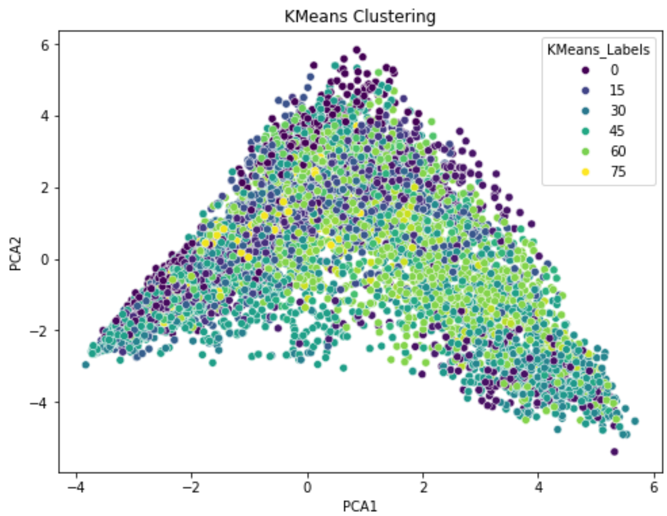

Fig. 8: KMeans Clustering

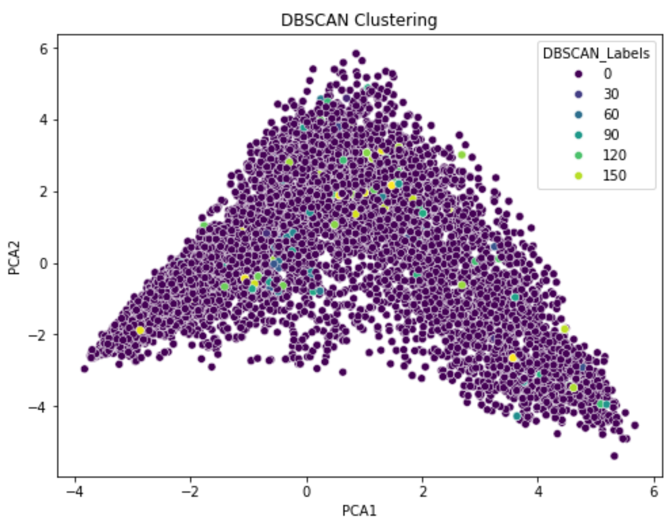

Fig. 9: DBSCAN Clustering

The K-means PCA principal components were 2 very common items: power treads and black king bar (BKB). This implies that the clustering algorithm mainly found the most common items and not any particular trend within an inventory.

DBSCAN PCA principal components had similar results. The 2 components were phase boots and blink dagger. This also implies that it just found very common items and formed clusters from there.

V. Conclusion

The analysis of player inventories in Dota 2 using unsupervised machine learning techniques such as association rule mining and clustering revealed interesting patterns and insights into player behavior. Association rule mining identified item combinations consistently used with specific heroes. Clustering, particularly with DBSCAN, demonstrated the potential to group data into meaningful clusters, which may correspond to hero-specific strategies or broader playstyles.

Despite these findings, limitations remain. The dataset's high dimensionality and the variability of itemization strategies present challenges in identifying universal patterns. Additionally, the exclusion of contextual factors, such as the current game version or opponents' heroes, restricts the analysis's depth. However, the results provide a foundation for understanding itemization trends and their relationship to gameplay strategies, offering valuable insights for future research and game analysis. Another piece of excluded information was the ranking of the player. Player's with less experience do not always pick the most optimal or predictable items for their match.

VI. Future Works

More information could be added to the dataset to allow association rule mining to see more complicated rules. In particular the current version of the game and the heroes that player is playing against can also influence that player's itemization. This would also introduce more dimensions into the data and could cause problems for other algorithms. The items' dimensionality could also be reduced if they were grouped into their respective groups. For example some items are purely offensive, purely defensive, and some have active abilities. Other items have no active abilities and only passive ones. The same could be done for the heroes.

References

[1] Kronusme, "Dota 2 API - Items Data," *GitHub Repository*. Available: https:\\github.com/kronusme/dota2-api/blob/master/data/items.json. [Accessed: Dec. 12, 2024].

[2] Kronusme, "Dota 2 API - Heroes Data," *GitHub Repository*. Available: https:\\github.com/kronusme/dota2-api/blob/master/data/heroes.json. [Accessed: Dec. 12, 2024].