



Combat Conundrum

Enhancing Dungeons & Dragons through
Machine Learning


Noah Waller, Ingrida Semenec, Alxandr Kane York,
Saswat Mishra, & Doug Stauffer

Dungeons & Dragons is a Massively Popular Tabletop Role-Playing Game

- D&D generates **\$100M - \$150M in revenue** and has an **player base of 50 million worldwide**
 - Players embody characters on a fantasy adventure
 - Player vs. Monster Combat
 - Combat encounters are managed by a Dungeon Master (DM)
-
- DMs use built-in "**Challenge Rating**" (CR) **system** to estimate combat difficulty



But the CR System is Not Effective...

←  r/DnD • 9 yr. ago
DM

[5e] Problem with Challenge Ratings

5th Edition

I've been running a 5e campaign for a while now, but have noticed some issues with the CR system. I've found this to be a common problem among DMs.


←  r/DnD • 6 yr. ago

Challenge rating sucks?

So I DM a campaign where the characters are level 6. During this session there were 4 party members

who were not meant to be characters that was

that combat like it

←  r/DnD • 2 yr. ago

My DM has discovered Challenge Rating and I hate his game now

5th Edition

So – we embarked on a quest to build a more effective model of D&D combat difficulty!

A D&D Dataset with Game-State Information

- D&D is often played online through chat applications like Discord
- Game-state information was stored via an in-game bot (Avrae) and collected into...

FIREBALL

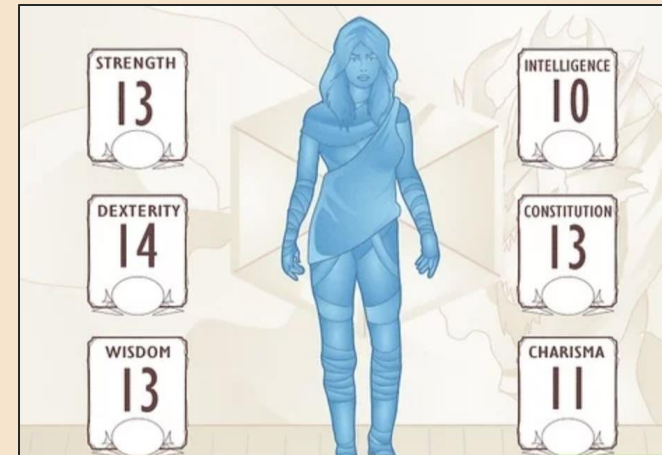
25,000 Unique Combat Sessions



3,000 Unique Players



Detailed Player Stats & Limited Monster Stats



Data Set Enigma

- ~25,000 JSONL files
- Data containing ~1000's of objects (compounding factors)
- Extreme investigation & avoiding noise was crucial



Character Class Defined

No Combat / Health Data

Anomalous Battle Data

Anomalous Battle / Party Data

Filtering Steps

24748

14824

13774

10842

Final Data Set



Silly Data, EDA is [NOT] for kids!

Here is an example of our long battle with our data!

25K JSON Files

data
1669403645-7aaa860...
1669406120-dd8c37e...
1669406338-1c8f5cd...
1669406592-85e20d5a-...
1669406672-6c18c26e-...
1669407116-03aa9506-...
1669407306-9f1295bf-4...
1669407643-495a7ebc-...
1669408207-0a00286b-...
1669408684-75e1b9c3-...
1669409601-f269d786-f...
1669409708-eed1a9bb-...

File example

The converted list: [['combat_id', 'event_type', 'timestamp', 'ontent', 'embeds', 'components', 'referenced_message_id', 'proba', 'command_name', 'called_by_alias', 'caster', 'targets', 'interac

Combine & Extract

```
def calculate_averages_for_encounters(encounters_dict):
    monster_levels = []
    player_healths = []

    for encounter_data in tqdm(encounters_dict.values(), desc="Calculating averages"):
        # Process monster levels
        if encounter_data['monsters']:
            total_level = sum(monster['level'] for monster in encounter_data['monsters'])
            monster_levels.append(total_level)
        else:
            monster_levels.append(0) # Append 0 if no monsters are present

    # Process player healths
    player_infos = encounter_data['player_info']
    total_max_health = sum(player['max_health'] for player in player_infos)
    player_healths.append(total_max_health)

    # Calculate overall averages
    monster_average = np.mean(monster_levels)
    player_average = np.mean(player_healths)

    return monster_levels, player_average
```

Filter

```
{
  'initial': 24748,
  'removal of standard classes': 18590,
  'removal of combat instances with no combat': 14824,
  'removal of combats with no health information': 13775,
  'removal of extreme pre-combat hp': 13774,
  'removal of extreme post-combat hp': 13774
}
```

Scale

```
# Function to calculate the weighted monster level
def weighted_monster_level(row):
    monster_number = row['monster_number']
    monster_level = row['monster_total_level']

    if monster_number == 1:
        multiplier = 1
    elif monster_number == 2:
        multiplier = 1.5
    elif 3 <= monster_number <= 6:
        multiplier = 2
    elif 7 <= monster_number <= 10:
        multiplier = 2.5
    elif 11 <= monster_number <= 14:
        multiplier = 3
    else: # 15 or more
        multiplier = 4

    return monster_level * multiplier
```

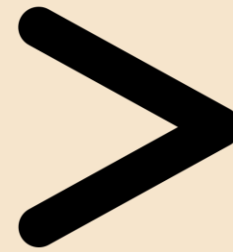
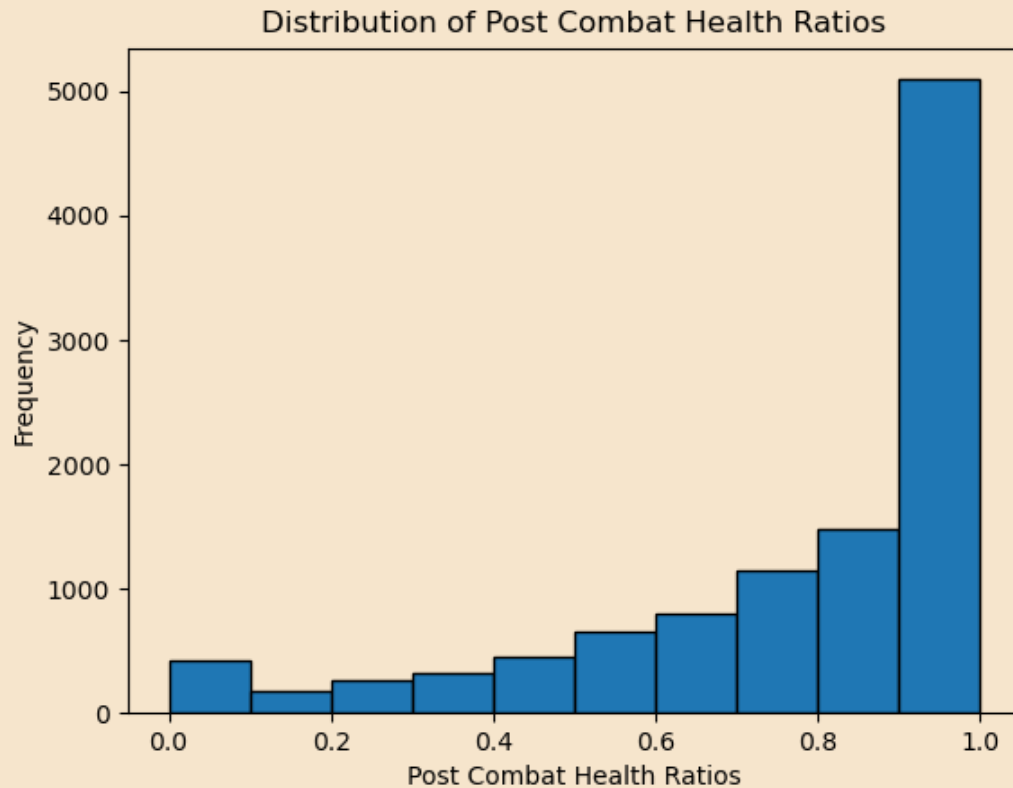
```
# Apply the function to create a new column
df['weighted_monster_level'] = df.apply(weighted_monster_level, axis=1)
```

Final Data Set!

player_individual_con...	player_individual_inte...	player_individual_wis...	player_individual_cha...	monster_types	monster_number	monster_total_level	party_total_level
[12]	[12]	[14]	[11]	['Warrior (level 4)']	1	4	10
[14]	[10]	[13]	[13]	['Flying Sword', 'Flyin...	3	2.5	5
[15, 16]	[13, 10]	[20, 16]	[12, 11]	['Elephant']	1	4	17
[16, 15]	[10, 13]	[16, 20]	[11, 12]	['Elephant', 'Magma M...	5	6	18
[16, 12, 14, 20, 15, 14...	[8, 10, 14, 10, 9, 12, 9...	[14, 13, 20, 14, 13, 19...	[17, 18, 12, 5, 16, 14, ...	['Death Dog']	1	1	58
[12, 14, 20, 16, 15, 14...	[10, 14, 10, 8, 9, 12, 1...	[13, 20, 14, 14, 13, 19...	[18, 12, 5, 17, 16, 14, ...	['Thug', 'Scout', 'Bandi...	7	2	58
[14, 16, 16, 15, 13]	[10, 14, 20, 12, 10]	[16, 13, 12, 15, 13]	[11, 8, 11, 18, 13]	['Couatl', 'Blink Dog', '...	3	4.5	16
[16, 13, 16]	[12, 9, 11]	[13, 12, 9]	[11, 18, 13]	['Blink Dog', 'Blink Do...	3	0.75	9
[12, 20, 14, 16, 15, 14...	[10, 10, 14, 8, 9, 12, 9...	[13, 14, 20, 14, 13, 19...	[18, 5, 12, 17, 16, 14, ...	['Black Pudding', 'Blac...	5	20	58
[16, 20]	[10, 15]	[16, 22]	[11, 11]	['Elephant']	1	4	29
[14, 14, 16]	[12, 14, 12]	[18, 10, 14]	[14, 16, 10]	['Three Earrings', 'Thr...	5	14	14
[19, 13, 18, 16, 20, 16]	[19, 12, 8, 19, 19, 8]	[12, 12, 12, 11, 12, 14]	[11, 15, 20, 20, 20, 18]	['Warhorse', 'Goristro', ...	7	22.5	63
[12, 20, 14, 20, 15, 16...	[10, 10, 14, 9, 11, 8, 1...	[13, 14, 20, 12, 13, 14...	[18, 5, 12, 5, 19, 17, 1...	['Imp', 'Imp', 'Imp', 'Im...	7	11	61
[19, 16]	[14, 10]	[16, 16]	[11, 11]	['Elephant', 'Elephant']	2	8	24
[15, 16]	[10, 10]	[14, 12]	[17, 9]	['Flying Sword', 'Winte...	2	3.25	17
[12, 20, 14, 15, 14, 20...	[10, 10, 14, 9, 12, 9, 1...	[13, 14, 20, 13, 19, 12...	[18, 5, 12, 16, 14, 5, 1...	['Giant Rat', 'Giant Ra...	5	0.5	62
[15, 15, 14, 18, 18, 13]	[16, 13, 11, 10, 11, 9]	[12, 17, 12, 11, 17, 12]	[10, 15, 19, 18, 8, 18]	['Werewolf', 'Werewolf...	8	15.75	18
[16, 14]	[10, 11]	[14, 16]	[20, 17]	['Ankylosaurus', 'Anky...	2	6	19
[6, 16]	[11, 9, 9, 12]	[12, 14, 16, 13]	[20, 16, 16, 11]	['Commoner']	1	0	23
[6, 14]	[9, 12, 9, 11]	[14, 13, 16, 12]	[16, 11, 16, 20]	['Intellect Devourer', 'I...	4	10	23
[6, 14]	[9, 12, 11]	[16, 13, 12]	[16, 11, 20]	['Mind Flayer', 'Intellec...	5	15	18
[6, 17]	[14, 9, 11, 12, 14]	[10, 20, 12, 13, 20]	[20, 14, 20, 10, 13]	['Griffon Cavalry Rider...	13	54	73
[6, 17]	[12, 14, 9, 11, 14]	[13, 10, 20, 12, 20]	[10, 20, 14, 20, 13]	['Death Knight', 'Deat...	3	39	73
[15]	[8]	[16]	[8]	['Tyrannosaurus Rex']	1	8	9
[15]	[8]	[16]	[8]	['Tyrannosaurus Rex']	1	8	9
[15]	[8]	[16]	[8]	['Beholder']	1	13	9
[16]	[8]	[16]	[8]	['Beholder']	1	13	9

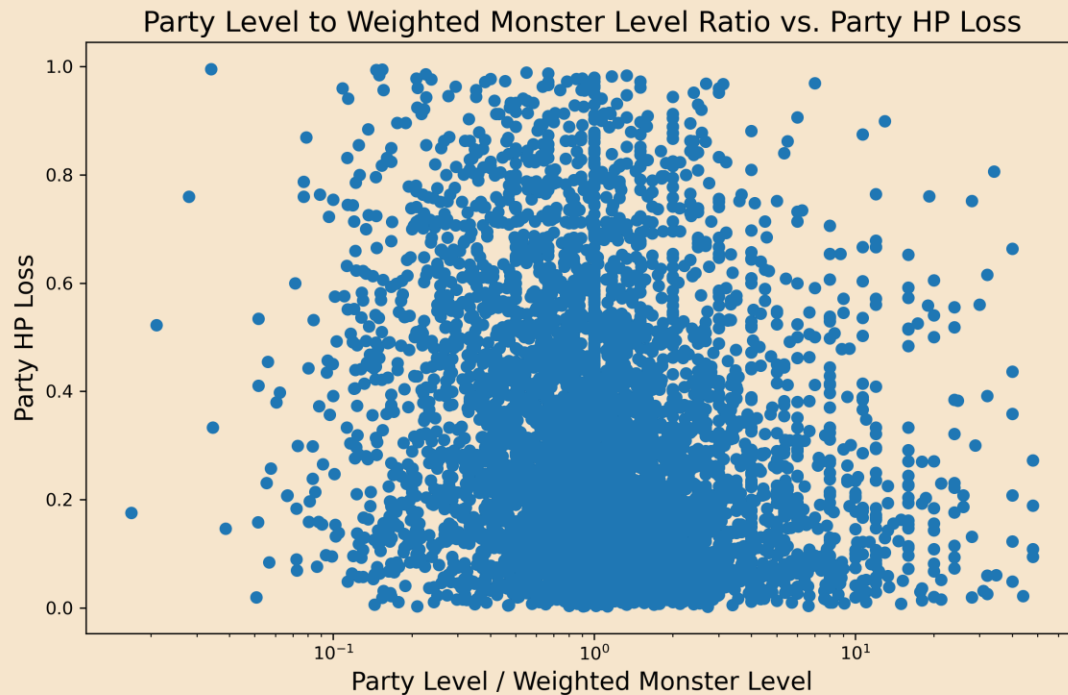
Imbalanced Data

- Number of total party kills (TPKs) was very low: 332
- Data was bias towards parties taking no damage



Challenge rating does not work!

proven by data



Success predictor

HP loss = pre-combat hp – post-combat hp / maximum hp



We need to consider other features!

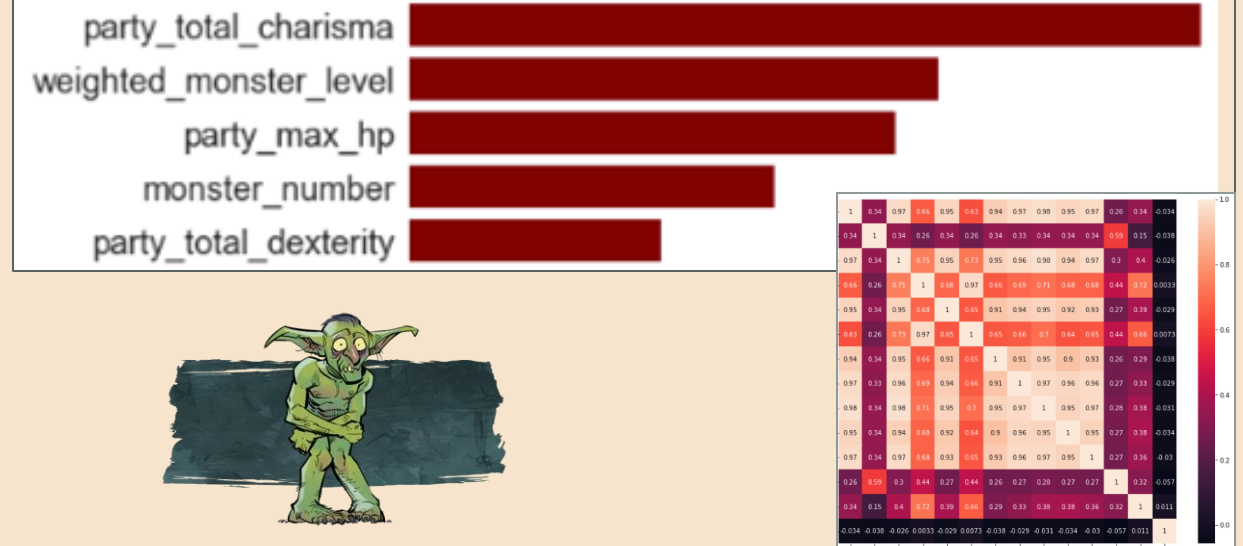
Oh! Behave Data!

Fit & Correlation Measuring

Model	Scoring		
	MSE	RMSE	R2
XGBoost	0.25	0.50	0.02
AdaBoost	1.65	1.28	-5.45
LightGBM	0.25	0.50	0.01
GradientBoost	0.26	0.51	0.00
PLA	0.25	0.50	0.01
OLS	0.48	0.69	-0.89
Random Forrest	0.27	0.52	-0.06

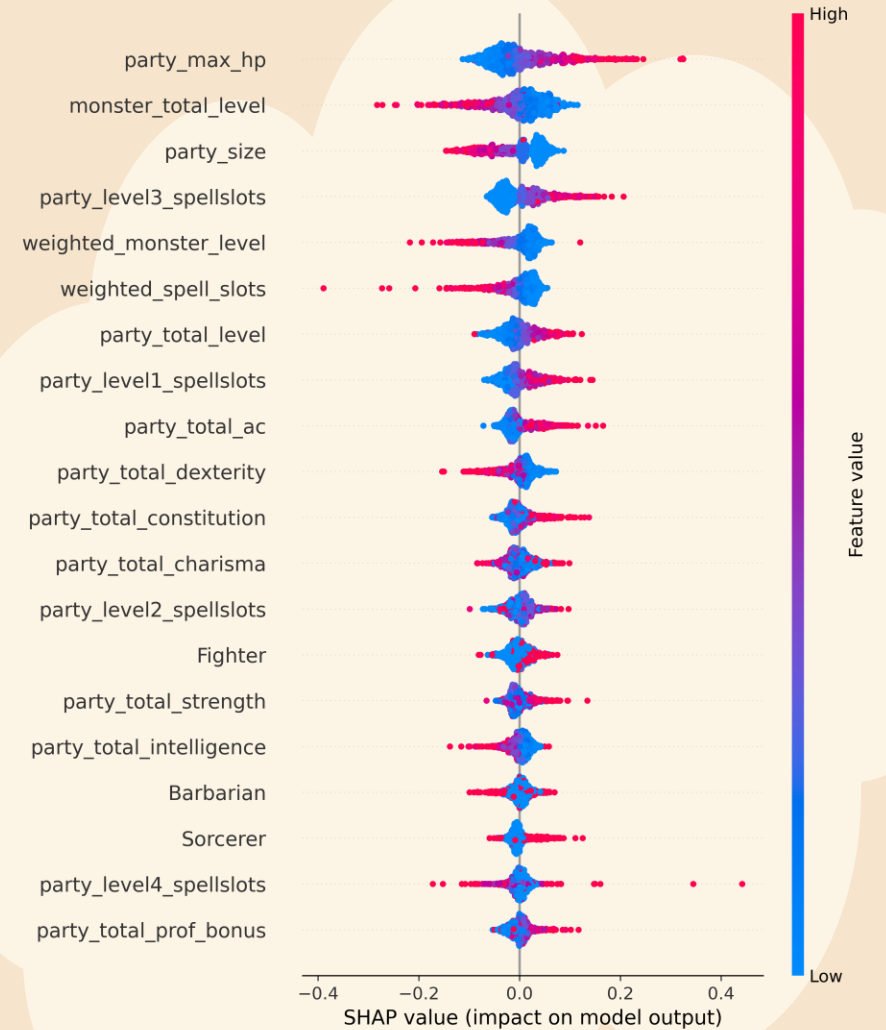
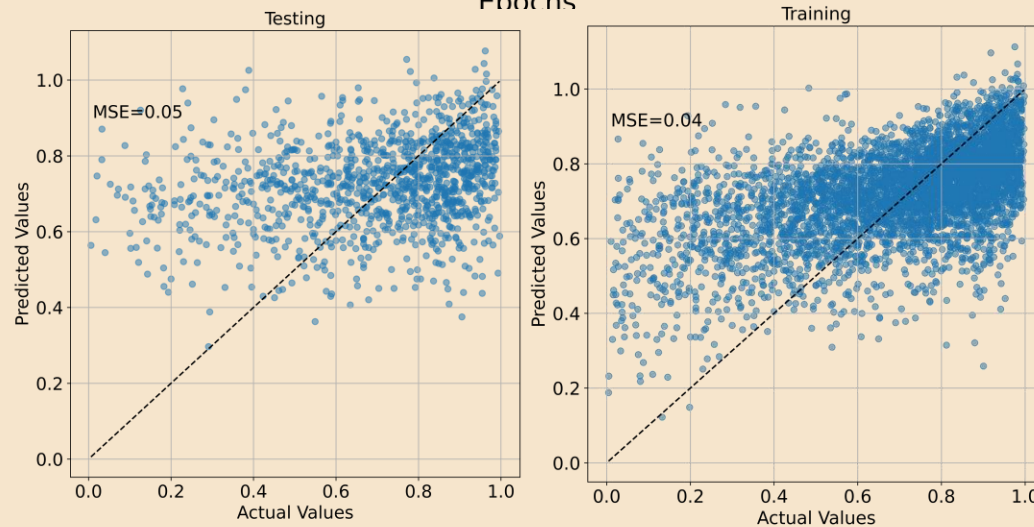
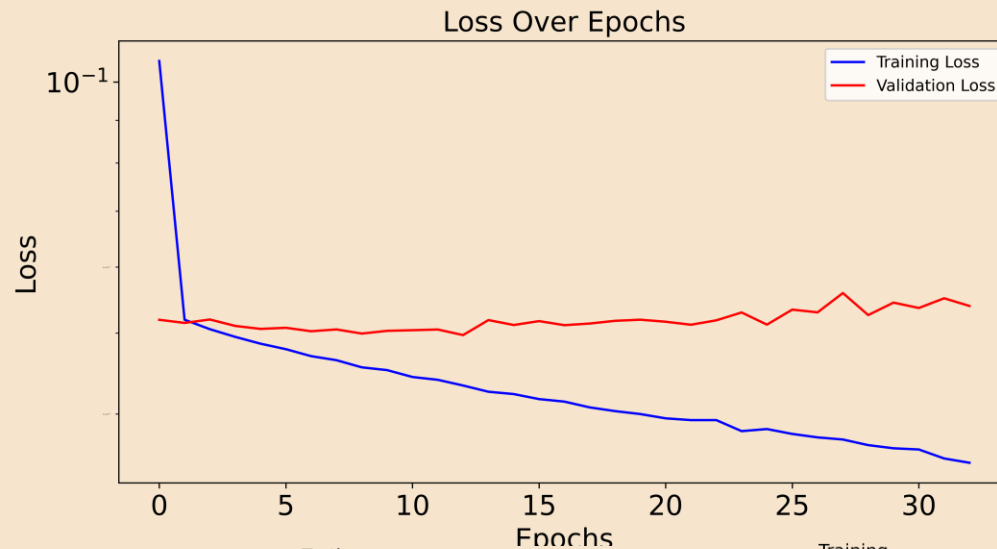
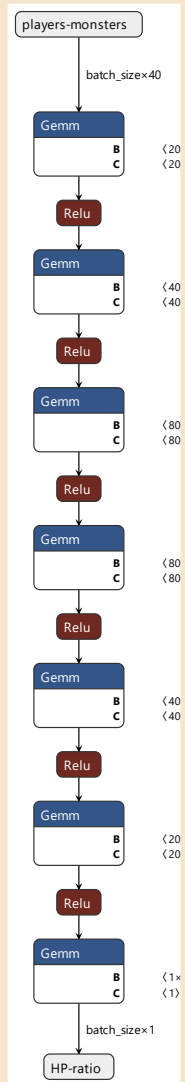
Target Score	~ 0 is best	0.2 >< 0.5	~ 1 is best
--------------	-------------	------------	-------------

Feature Importance & Correlation

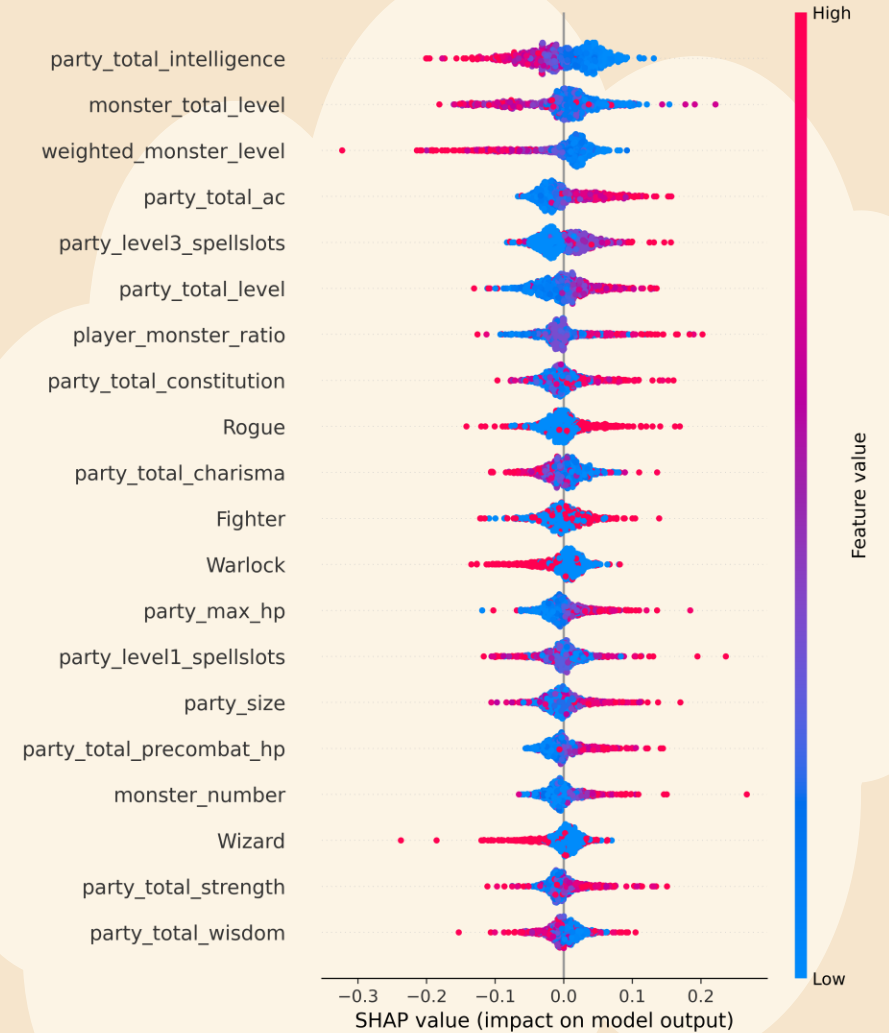
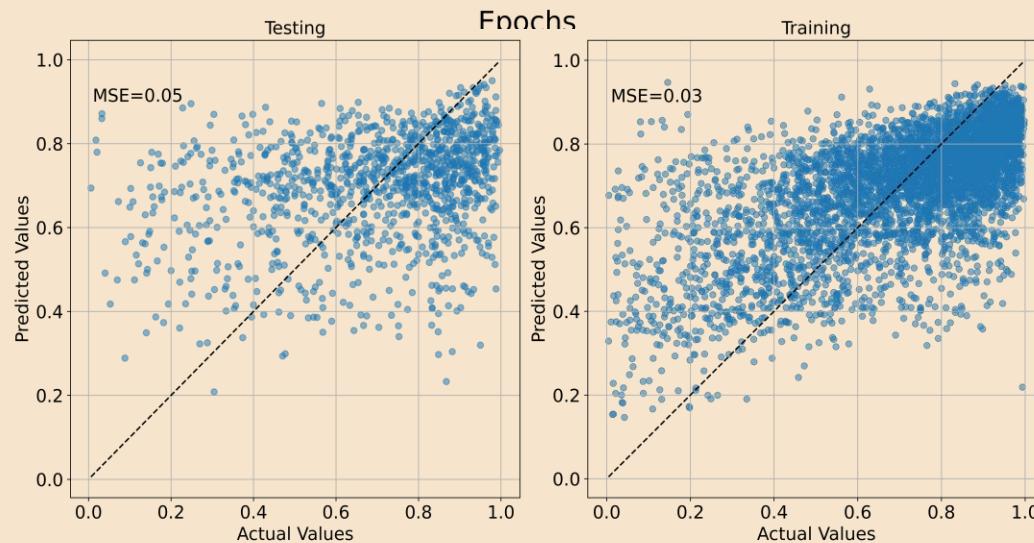
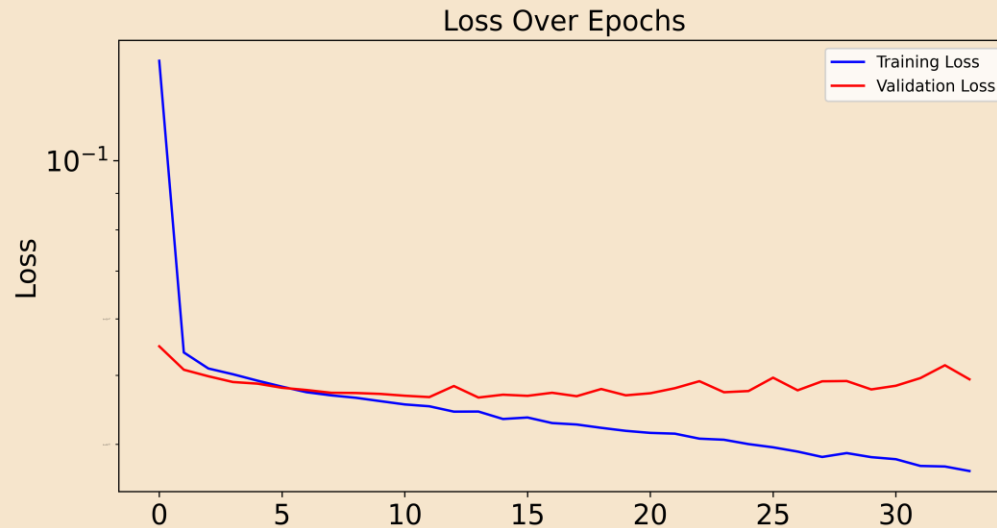
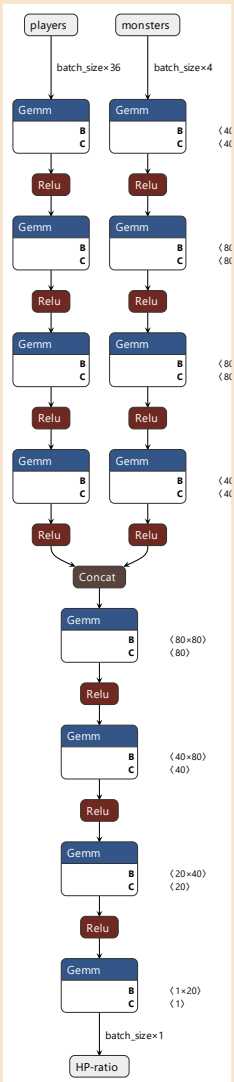


- Models resulted in overfitting & low correlation with features used to the targeted prediction
- Feature variable importance to each model was random and extremely low

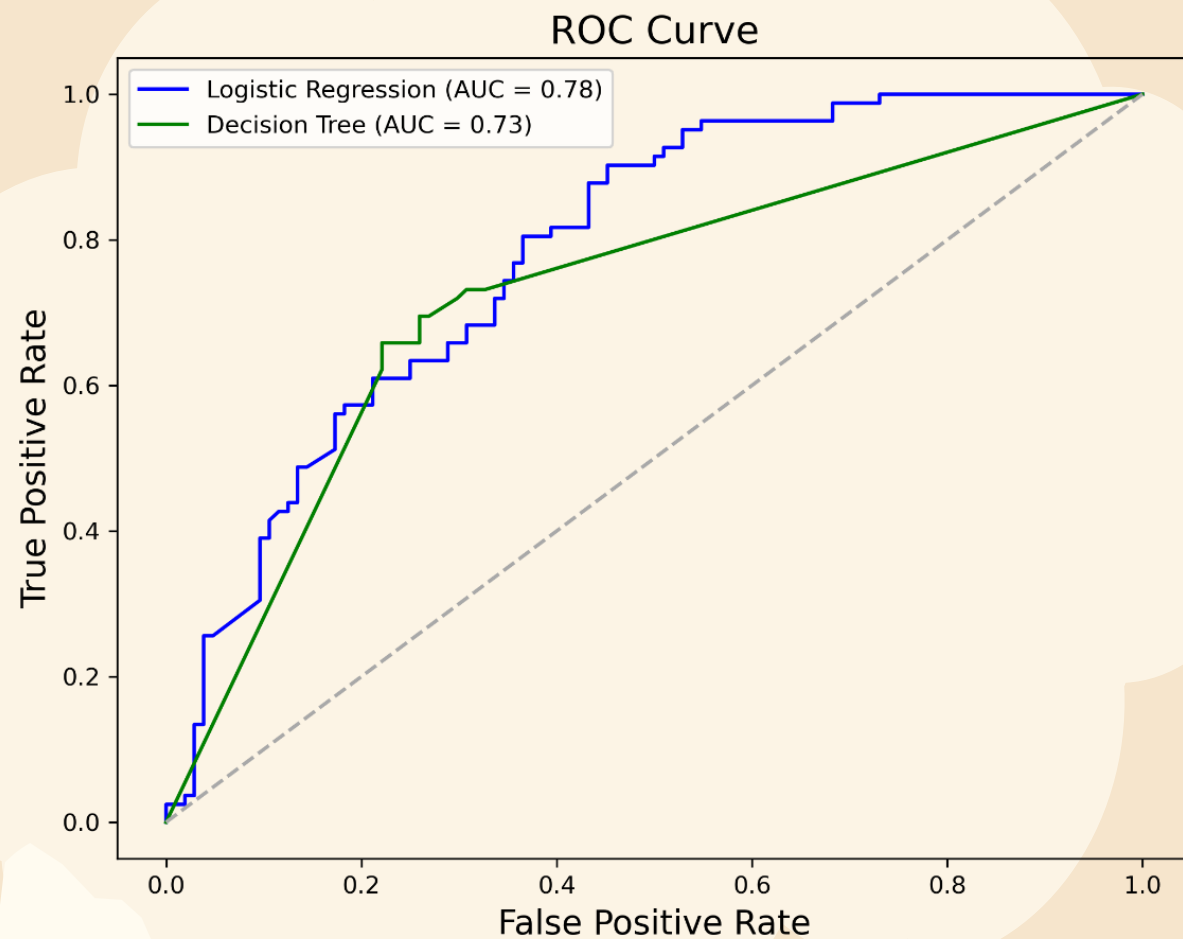
Deep Neural Network



Deep Neural Network






Classifier: Total Party Kill vs. Survival



LR

DT

	Precision	Recall
	0.86	0.57
	0.62	0.88
	0.75	0.74
	0.68	0.70

Outcomes and Future Directions



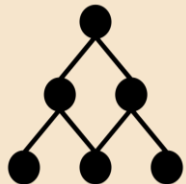
A closed study performed by Wizards of the Coast via the *Avrae* bot with stricter parameters (no homebrew content)



Inventory analysis (magical items)




More less successful combats




Hierarchical models including individual Dungeon Masters as their own levels

Graphic User Interface Can Provide DMs with Accessible Tool



D&D Enhanced Combat Difficulty Predictor



Player Information

Party Size

Party Total Level

Party Total Prof. Bonus

Party Total AC

Monster Information

Monster Number

Monster Total CR

>

No HP Loss → TPK



Thank You!