Project 4 Proposal Document

Predicting Monster Encounter Outcomes in Dungeons and Dragons

Our group wants to create a machine learning model that can determine whether a party of adventurers will successfully defeat a monster in an encounter based on the monster's attributes, such as challenge rating (CR), hit points (HP), and abilities.

Datasets:  
-<https://www.kaggle.com/datasets/shadowtime2000/dungeons-dragons?select=monsters.csv>  
-<https://www.kaggle.com/datasets/poketch/dungeons-dragons-5e-monster-data?select=5e_monster_data_5eTools.csv>

What is contained in the dataset:

* **Name**: The name of the monster.
* **Size**: Monster size (e.g., Medium, Large, etc.).
* **Type**: Monster type (e.g., Dragon, Undead, Beast).
* **Alignment**: Monster’s moral alignment (e.g., Chaotic Evil, Neutral Good).
* **Challenge Rating (CR)**: Represents the combat difficulty of the monster.
* **Armor Class (AC)**: Monster’s defense rating.
* **Hit Points (HP)**: The health points of the monster.
* **Attributes**: Strength, Dexterity, Constitution, Intelligence, Wisdom, and Charisma.
* **Speed**: Movement speed of the monster in combat.
* **Damage Resistances/Immunities**: Specific resistances, which could impact combat outcomes.
* **Legendary Actions**: Special actions that some monsters can take, increasing their combat difficulty.

The goal:  
-Predict the likelihood of a monster being defeated in combat.

Target Variable:

* Assign a binary label (Defeated: Yes/No) based on the monster’s Challenge Rating (CR) and assumed player party level. For example:
  + Monsters with a CR much higher than the party level are more likely to win.
  + Monsters with a CR lower than the party level are more likely to lose.

How its done:  
1. **Generate Synthetic Data for Combat Outcomes**

Add a new column to the dataset to simulate combat outcomes:

* Define a party level (e.g., Level 5 adventurers).
* Assume monsters with a CR <= party level are likely to be defeated, and monsters with a CR > party level are likely to win.
* Add some randomness to simulate real encounters.

2**. Exploratory Data Analysis (EDA)**

Use **Pandas**, **Matplotlib**, and **Seaborn** to explore the data:

* **Distribution of Challenge Ratings (CR):**
  + Visualize the frequency of monsters across different CRs.
  + Plot survival rates for each CR.
* **Monster Types vs. Defeat Rate:**
  + Analyze which types of monsters are most or least likely to be defeated.
* **Key Stats vs. Combat Outcomes:**
  + Plot correlations between stats like HP, AC, and survival.

3.  **Machine Learning Model**

Use **Scikit-learn** to predict whether a monster will be defeated based on its attributes.

* **Target Variable**: Defeated (Yes/No).
* **Features**:
  + Challenge Rating (CR).
  + Armor Class (AC).
  + Hit Points (HP).
  + Size (encoded numerically).
  + Attributes (Strength, Dexterity, etc.).
  + Type (one-hot encoded).
  + Damage Resistances/Immunities (aggregate score).

Steps:

-Split the data into training and testing sets

-Train models

* + Logistic Regression for baseline.
  + Random Forest or Gradient Boosting for better performance.

-Evaluate the model

Visualizations:

Create **Plotly** visualizations for interactive analysis:

**Monster Stats vs. Predicted Outcome**:

* + Scatter plot comparing CR and HP against the likelihood of defeat.

**Feature Importance**:

* + Bar chart showing which features (e.g., AC, HP) are most important in predicting combat outcomes.

Deployment (if time allows):

* **Interactive Encounter Predictor**: Build a **Flask** where users input monster stats to predict combat outcomes.
  + Example inputs: CR, HP, AC, etc.
  + Output: Predicted outcome (Yes/No) and probability.
* **Web Dashboard**: Use **Tableau** to create a dashboard visualizing:
  + Monster likelihood of defeat probabilities.
  + Stats vs. outcomes.

### **Timeline (2 Weeks)**

| **Week** | **Task** |
| --- | --- |
| Week 1 | Dataset cleaning, EDA, and feature engineering. |
| Week 2 | Model building, evaluation, visualization, and deployment. |