Part 2: Predicting Troop Betrayal in the War Against the Phrygians

Objective

Construct a predictive model which identifies those members of an army that are most likely to be traitors and defect to the enemy during a war with the Phrygians. Using the factors like Greed , temptation,poor etc.

Approach Overview

- The task was approached using a multi-step process that involved:

1. **Hypothesis Formation**: Identifying key factors that could influence a soldier's likelihood to betray.
2. **Feature Engineering**: Quantifying these factors into features that could be used as inputs to a machine learning model.
3. **Model Design**: Designing a decision flow system of a which can identify the soldier's likelihood of betray.
4. **Evaluation and Scalability Considerations**: Considering how the system can adapt to new data and become more effective over time

1. Hypothesis Formation

- **1. Greed (x1)**

  It is a parameter to count the probability of betrayal scaled between 1 to 10

- Respect for commanders:(x2) scaled between 1 to 10
- Financial Debt :(x3) scaled between 1 to 10
- Past loyalty :(x4) scaled between 1 for loyal and 0 for betray
- Temptation by Enemy :(x5) scaled between 1 to 10
- Missions:(x6) number of mission done
- Family hardship :(x7) binary scaled 0 for no 1 for yes.
- Military performance : scaled between 1 to 10

Interaction Features and Transformations

- **1. Greed × Financial Debt (x1*x3)**
- **2. Respect × Loyalty (x2*x4)**

3. Squared Morale (x2*x2)

4. Logarithmic Transformation of Debt (log(x3+1))

1. Recent Morale Drop: Capture the rate of change in morale by looking at recent morale scores compared to historical averages

$$\text{MORALE DROP} = \frac{\text{Morale(Current)}}{\text{Average Morale(Past)}} - 1$$

A significant drop in morale may be a stronger indicator of betrayal risk than the absolute morale value.

2.Cumulative Exposure to Enemy: Track cumulative exposure to enemy offers over time to capture long-term temptation effects.

- Cumulative Temptation=$\sum x2(t)$
- **Temporal Features**:
- **Time Since Last Betrayal in Squad**: This feature can indicate whether betrayal is recent in a soldier's squad. The longer the time since the last betrayal, the lower the likelihood of new betrayals.

$$\text{Time\_Since\_Last\_Betrayal} = \frac{1}{\text{Days since last betrayal in squad}}$$

3.Mission Risk to Morale Ratio: High mission risk combined with low morale could increase the risk of betrayal.

$$\text{Risk\_Morale\_Ratio} = \frac{\text{Mission Risk}}{\text{Morale}}$$

Rolling Average Performance: Use rolling averages of the soldier's performance score over time to smooth out performance trends. This captures recent performance trends better than a single performance score.

$$\text{Rolling Avg Performance} = \frac{1}{n} \sum_{i=t-n}^{t} \text{Performance}_i$$

Conclusion for selection

Each feature was chosen based on psychological, financial, and behavioral factors that contribute to a soldier's likelihood of betrayal. Features like **Greed**, **Respect**, and **Financial Debt** quantify personal motivations, while **Past Loyalty Record** and **Military Performance** offer insights into past behavior and future predictions. Interaction features like **Greed × Debt** and **Respect × Loyalty** combine multiple high-risk factors to enhance the model's predictive power.

Through these carefully chosen features and transformations, the model becomes more robust in identifying traitorous tendencies, allowing for early detection and prevention of betrayal.

XG BOOST(Model Used)

- The formula in XGBoost looks like:

$$P(\text{Betrayal}) = \sigma\left(\sum_{k=1}^{K} f_k(x)\right)$$

- Where $f_k(x)$ represents the decision tree built at the k-th iteration. The input to each tree is the set of features {x1,x2,...,x8 and the engineered features x1×x3 and log(x3+1).

Mathematical Representation in XGBoost

- **Tree Splitting Criteria**: The splits in the tree are determined by finding the best threshold for each feature to minimize the loss function, usually based on a **gradient boosting algorithm**.
- For example:

1. Is greed x1>7?
2. Is debt x3>8?
3. Is mission count x6<3?

- **Ensemble Prediction**: Each tree contributes an additive score. The final prediction is based on the summation of the contributions from all trees:

$$\hat{y} = f_1(x) + f_2(x) + f_3(x) + \cdots + f_K(x)$$

Where fk(x) denotes the prediction of kth-node

Accuracy

$$\text{Accuracy} = \frac{number\ of\ positive\ prediction}{total\ size\ of\ the\ sample\ space}$$

Accuracy is also defined as -

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Where

TP-> correctly predicted positive cases

TN -> correctly predicted negative cases

FP -> incorrectly predicted positive cases

FN-> incorrectly predicted negative case

Conclusion

- By using XGBoost for predicting system aggression, we achieve a stable capacity for large-scale data, with high throughput of decision-making in the changing battlefield landscape; With periodic model updates, dynamic feature engineering the system is able to rank-order soldiers by the betrayal parameters