Joann Guyoncour

MAIS 202 – Deliverable 2

Project: Clash Royale Win Prediction Model

1. Problem statement

The goal of this project is to predict the winner between two Clash Royale players based solely on the composition of their decks. Players often know that certain decks perform better against others, and I aim to determine if a model can accurately predict the outcome based purely on deck composition, without considering the players' skill levels. The central question is whether deck composition alone can be a reliable predictor of victory, independent of player skill.

2. Data preprocessing

I will work with a slightly different dataset than previously (Clash Royale S18 ladder datasets), as this one includes data on the players' remaining hit points (HP) and the number of crowns. This allows me to calculate a win margin, which will help the model adjust its weights more accurately during backpropagation. Additionally, this dataset contains 38 million matches, a reduction from the original 480 million, which should be sufficient given the limited capacity of my computer.

Many features in the dataset are irrelevant for predicting the winner (e.g., trophy count, match duration, etc.). During preprocessing, I reduced the dataset to 22 features: 16 categorical features representing the card IDs of the 8 cards in each player's deck, and 2 numerical features representing the total card level and the average elixir cost of each player. I also created a 'winner_value' column, which takes a value of 0 or 1, indicating whether player 1 or player 2 won the match. Since the dataset originally labeled the winner as player 1, I shuffled the data to ensure that the winner could be either player 1 or player 2.
To compute the win margin, I used a weighted average of the normalized crown_difference (weight = 0.7) and HP_difference (weight = 0.3), scaled to a range between 0 and 1 using the sigmoid function.

3. Machine learning model

The task is a binary classification problem (player 1 wins (0) or player 2 wins (1)), so a confusion matrix will be used for model evaluation. I started with logistic regression as a baseline model due to its simplicity and ease of implementation. However, I do not expect high accuracy due to the large number of features (22) and the likely non-linear relationships between them.
Categorical features (e.g., card.id) were one-hot encoded, while numerical features (e.g., average.elixir and total.level) were normalized using the scikit-learn functions OneHotEncoder() and StandardScaler().

4. Preliminary results

These are the preliminary results obtain from logistic regression:

Accuracy: 0.5883054486869326

Joann Guyoncour

Confusion Matrix:
[[111784 79490]
 [77921 113154]]

The accuracy is slightly better than random guessing. I also trained the model using only the numerical features (total.card.level and average.elixir), resulting in a similar accuracy (0.57). Training the model using only the card composition led to poorer results (accuracy = 0.52). This suggests that the total.card.level feature provides the most relevant information, indicating that players with higher-level cards (and thus better stats) have a better chance of winning. The composition of the deck itself does not seem to contribute significantly to predicting the winner in the logistic regression model. The limited sample size (2,000,000 matches) may be a factor, but the primary issue is likely the non-linearity of the relationship between the cards and how they have been encoded (using one-hot encoding).

5. Next steps

Since logistic regression is not satisfactory as expected, I'm going to try implementing a Forward Neural Network and convert card.id into a vector representation better integrating the relationship between the cards.
If the model achieves satisfactory accuracy, I plan to make it accessible through a web interface, where users can input two Clash Royale decks and receive a predicted win probability.