Clash Royale Match Outcome Prediction

Faraa Awoyemi, Lilia Benabdallah, Ilyes Ben Younes, Lea Hadj-Said

1 Introduction

The goal of this project is to predict the result of a 1v1 Clash Royale match using only the decks of both players. We work on a supervised classification problem where the model must predict if Player 1 wins or loses the match.

We found our datasets on Kaggle. The first dataset contains real competitive matches played on the top ladder. It is not a tournament dataset. Each row corresponds to one independent match between two players. The second dataset contains information about the cards such as elixir cost, rarity, win rate and usage rate.

At first, we only use deck composition. Then, in Phase 2, we add more strategic information to improve the predictions.

2 Dataset Description

The main dataset contains 2311 matches. For each match, we have the 8 cards of Player 1, the 8 cards of Player 2, and the number of crowns scored by both players. We also add a new column called `winner` or `win_p1` which is equal to 1 if Player 1 has more crowns than Player 2, and 0 otherwise.

This dataset is a collection of matches played by different players at high level. The dataset is naturally imbalanced. Player 1 wins around 72% of the matches. This happens because Player 1 is often the player who is higher on the ladder or in a winning streak. This imbalance reflects real game conditions and should not be artificially corrected.

3 Problem Formalization

Let $X \in R^n$ be the feature vector representing a Clash Royale match, where the features correspond to the encoded cards of both players and the aggregated deck statistics. Let $y \in \{0, 1\}$ be the target variable, where $y = 1$ means that Player 1 wins the match, and $y = 0$ otherwise.

Our objective is to learn a classification function $f : X \to y$ that predicts the outcome of a match given the two decks. This problem is therefore a supervised binary classification task.

4 Exploratory Data Analysis

We start by checking the quality of the dataset. There are no missing values, no duplicated rows, and all crown values are valid between 0 and 3.

4.1 Most Played Cards

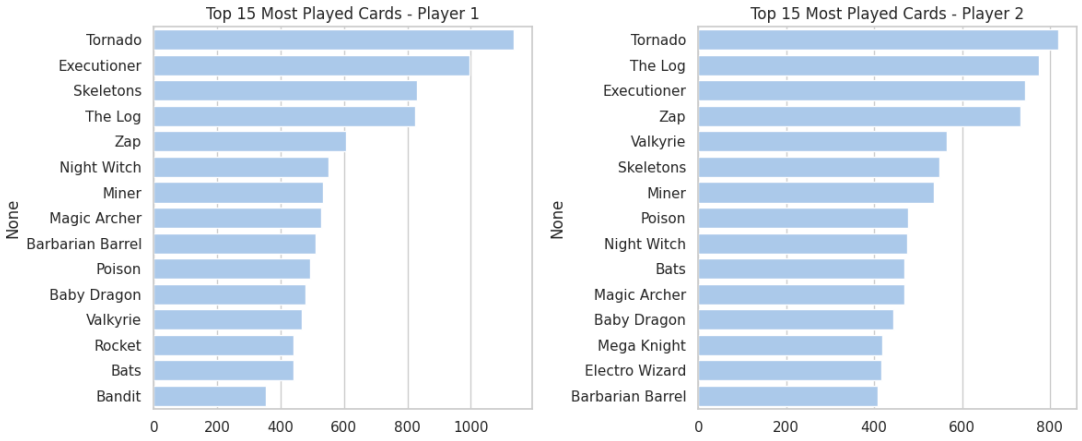

Figure 1: Top 15 Most Played Cards for Player 1 and Player 2

We observe that some cards like Tornado, Executioner, The Log and Zap are very common in both decks. This means that these cards are part of the current meta of the game, meaning they are considered strong, versatile, and widely used by high-level players. As a result, the dataset is not balanced between all cards: some cards appear very frequently while others are much rarer.

This has an important impact on machine learning. The models will naturally learn better the behaviour of popular cards because they appear in many matches. On the contrary, rare cards bring less information to the model and are harder to learn correctly. This creates a natural bias in the dataset: predictions are more reliable for decks using popular cards and less reliable for uncommon strategies.

This observation also shows that the model is learning patterns that are strongly linked to the current game meta, and not only to pure card strength.

4.2 Match Outcome Distribution

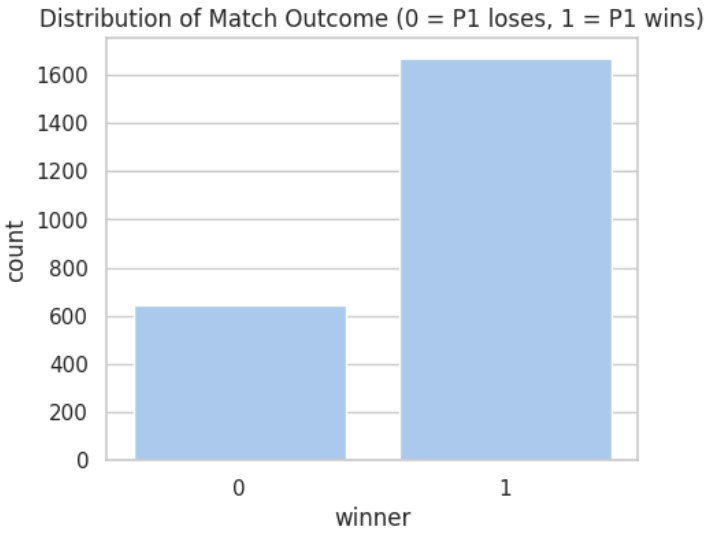

Figure 2: Distribution of Match Outcome

We clearly observe a strong class imbalance in this dataset. Player 1 wins around 72% of the matches, while Player 2 only wins about 28%. This imbalance does not come from an error in the data, but from the way the dataset was collected. The matches come from the top ladder, where Player 1 is often the player with the highest rank or the one currently on a winning streak. As a result, Player 1 naturally has a higher probability of winning.

We decided not to correct this imbalance using techniques such as undersampling or oversampling. Our goal is to stay as close as possible to real game conditions. Artificially balancing the dataset could create unrealistic situations and bias the interpretation of the results. However, this imbalance must be taken into account when analyzing model performance, especially for metrics related to the losing class.

This class imbalance problem is well known in machine learning. As shown by He and Garcia (2009), standard classification algorithms tend to be biased toward the majority class, which leads to poor detection of the minority class. This directly explains the low recall observed for Player 1 losses in our models.

5 Data Preprocessing

We apply One-Hot Encoding on the 16 card slots. After encoding, the dataset has about 1300 binary features.

We split the dataset into training and testing sets with an 80/20 ratio using stratification.

6 Phase 1: Baseline Models

We train Logistic Regression, Decision Tree, and KNN. To evaluate the performance of each baseline model, we first report quantitative metrics, and then analyze their confusion matrices on the test set.

6.1 Quantitative Evaluation

Model	Accuracy	Precision	Recall (0)	F1-score	AUC
Logistic Regression	0.63	0.77	0.46	0.73	0.57
Decision Tree	0.62	0.71	0.17	0.75	0.50
KNN	0.73	0.73	0.05	0.84	0.54

Table 1: Performance of baseline models (Phase 1)

This table highlights the limitations of the baseline models. KNN achieves the highest accuracy (0.73), but its recall on the losing class is extremely low (0.05), meaning that it almost never detects Player 1 defeats. This confirms that KNN mainly predicts the majority class. Decision Tree also performs poorly on the minority class, with a recall of only 0.17, showing an unstable behavior. Logistic Regression appears to be the most balanced baseline model, with a significantly higher recall on defeats (0.46), even if its accuracy is slightly lower. Overall, this confirms that accuracy alone is misleading in this highly imbalanced context.

6.2 Confusion Matrix Analysis

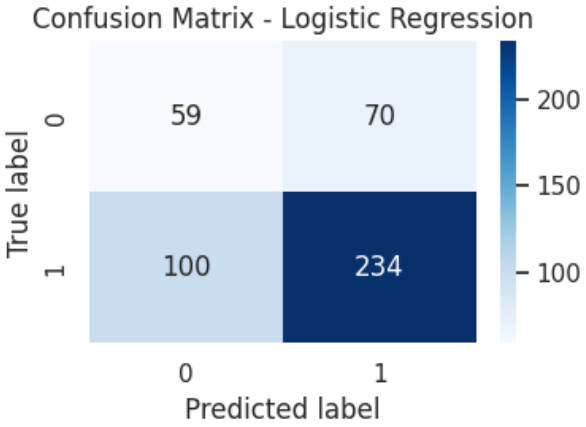

Figure 3: Logistic Regression Confusion Matrix

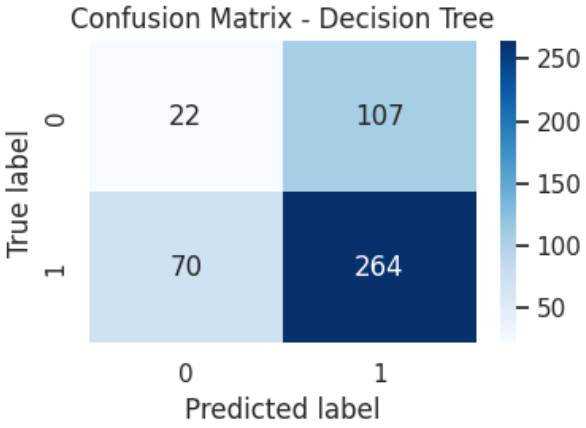

Figure 4: Decision Tree Confusion Matrix

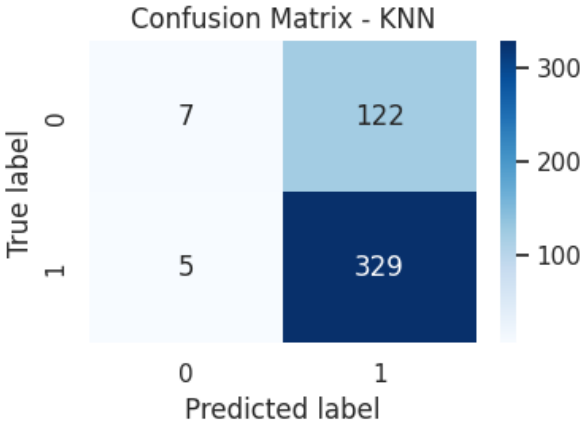

Figure 5: KNN Confusion Matrix

From the confusion matrices, we can clearly observe the behavior of each baseline model. For Logistic Regression, the model correctly predicts a large part of Player 1 victories, but it still

makes many mistakes on defeats. A significant number of losses are predicted as wins. This confirms that the model is influenced by the strong class imbalance of the dataset. However, it remains the most balanced model of Phase 1, with a moderate accuracy and the best compromise between precision and recall.

For the Decision Tree, the situation is worse for the losing class. The model predicts very few defeats correctly and misclassifies most losses as victories. This results in a low recall for the losing class, even if the overall accuracy remains around 0.62. This shows that the tree mainly follows the global tendency of the dataset and does not manage to learn precise decision rules.

The KNN model shows the strongest bias. It almost always predicts that Player 1 wins. As a result, it has a very high recall for victories but an extremely low recall for defeats. This means that KNN mainly copies the majority class and completely fails at detecting losses.

Overall, these confusion matrices confirm that all baseline models struggle to correctly predict defeats because of the strong class imbalance. Even if some models reach acceptable accuracy values, this accuracy is misleading because it mainly reflects the dominance of victories in the dataset. Among the three models, Logistic Regression remains the most reliable one in Phase 1, even if its performance is still limited.

7 Phase 2: Dataset Enrichment

In Phase 1, we only used the deck composition to predict the match outcome. However, this representation is very limited. Two decks can contain very different types of cards in terms of elixir cost, rarity or strength, but the model does not see this information directly. For this reason, we decided to enrich our dataset using an additional external dataset.

The second dataset was also taken from Kaggle. It contains global statistics for each Clash Royale card, such as elixir cost, card rarity, average win rate and usage rate. This dataset does not describe individual matches, but general properties of the cards based on a large number of games.

We merged this second dataset with our main match dataset by linking each card name to its corresponding statistics. This operation was done for all card slots of both players. After merging, some values were missing because not all cards were present in the statistics dataset. These missing numerical values were filled using the mean of each column to avoid losing data.

The goal of this Phase 2 is to give more strategic information to the models. Instead of only knowing which cards are played, the models now also know if a deck is expensive or cheap, common or rare, and globally strong or weak according to statistics. This allows the models to learn higher-level patterns about deck strength and play style.

8 Advanced Feature Engineering

We compute average deck elixir, win rate, usage rate and rarity.

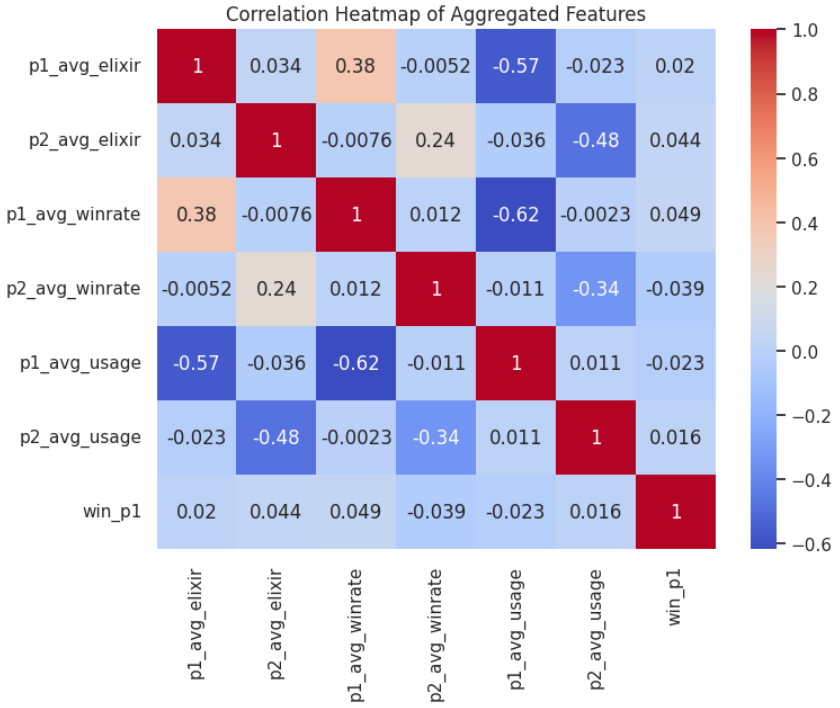

Figure 6: Correlation Heatmap of Aggregated Features

We observe that the correlations between the aggregated features and the target variable are globally weak. This means that no single feature is sufficient to clearly explain the match outcome by itself.

Some moderate correlations appear between the average elixir cost and the average win rate of Player 1. This shows that decks with a higher elixir cost can sometimes be associated with stronger cards and better global performance. However, this does not guarantee a victory.

We also observe negative correlations between the average usage rate and the average win rate. This suggests that very popular cards are not always the most efficient for winning, and that some less frequent cards can still be very strong in the right strategies.

Finally, the weak direct correlation with the match outcome confirms that predicting a Clash Royale match is a complex task. The result does not depend on only one deck feature, but on the combination of multiple factors and on in-game decisions that are not present in the dataset.

9 Advanced Models

We train Random Forest, XGBoost, and CatBoost as advanced classification models.

Random Forest is an ensemble learning method based on the combination of multiple decision trees. Each tree is trained on a random subset of the data and features, which allows the model to reduce overfitting and improve generalization compared to a single decision tree.

XGBoost is a state-of-the-art boosting algorithm widely used for structured data classification tasks. According to Chen and Guestrin (2016), its efficiency comes from optimized tree learning, regularization, and parallel computation, which makes it particularly suited for large-scale and complex datasets.

CatBoost is another gradient boosting algorithm designed to efficiently handle categorical features and reduce overfitting.

We report below the results obtained with the three advanced models.

Model	Accuracy	Precision	Recall (0)	F1-score	AUC
Random Forest	0.72	0.74	0.11	0.83	0.56
XGBoost	0.72	0.74	0.16	0.83	0.59
CatBoost	0.73	0.74	0.12	0.84	0.60

Table 2: Performance of advanced models (Phase 2)

This table shows that the advanced models achieve slightly better performance than the baseline models. CatBoost obtains the highest accuracy (0.73) and the highest AUC (0.60), confirming its better overall discriminative ability. However, the recall on the losing class remains low for all three models, which indicates that predicting Player 1 defeats is still difficult despite the dataset enrichment. XGBoost presents a slightly better recall on the minority class (0.16), while Random Forest remains the weakest in this regard. Overall, the improvements brought by Phase 2 are real but remain limited by the strong class imbalance and the lack of in-game information.

To further analyze the error distribution of the best performing model, we now examine the confusion matrix of CatBoost.

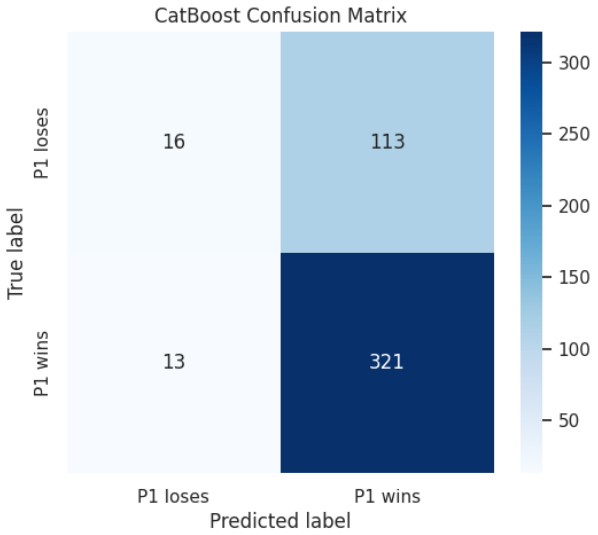

Figure 7: CatBoost Confusion Matrix

The confusion matrix of CatBoost confirms the tendencies observed in the quantitative evaluation. Most prediction errors still correspond to Player 1 losses being predicted as wins, which reflects the strong class imbalance of the dataset. Although CatBoost improves the overall discrimination performance, it still struggles to correctly identify the minority class.

CatBoost was introduced by Dorogush et al. (2018) to specifically handle categorical features while reducing prediction shift and overfitting through ordered target statistics. This property explains why CatBoost performs slightly better than classical boosting methods in our experiments.

Even if CatBoost is the best model, the improvement compared to Phase 1 remains limited. The confusion matrix of CatBoost shows that most errors still come from the difficulty to correctly predict Player 1 losses. The model still tends to favor the majority class.

These results confirm that adding better features improves the model, but the prediction task remains difficult due to the strong class imbalance and the absence of real gameplay data.

10 Discussion

Even after adding many new features and using more advanced models, the global performance of our models remains limited. The best accuracy we obtain is around 0.73 with CatBoost and the voting ensemble. This result is only slightly better than the baseline accuracy of the dataset, which is around 72% because Player 1 wins most of the matches.

This clearly shows that the dataset is highly imbalanced. Most models learn that predicting "Player 1 wins" is often enough to obtain a good accuracy. This explains why some models, like KNN, mainly predict victories and almost never predict defeats. As a consequence, the recall for defeats is very low.

Another major limitation is the lack of real gameplay information. We only use deck composition and general card statistics. Important elements such as card levels, player skill, timing, placements, and elixir management during the match are completely missing. These elements have a very strong impact on the result of a match.

We also observe that even when two players use very similar decks, the outcome can be totally different depending on how the match is played. This explains why the models struggle to learn strong and stable decision rules.

Finally, adding more complex models does not automatically guarantee much better results. Even boosting models like XGBoost and CatBoost remain limited by the quality and the type of available data.

11 Conclusion

In this project, we studied the problem of predicting the outcome of a Clash Royale 1v1 match using machine learning. We started with a dataset containing only the decks of both players and the final number of crowns. We built a binary classification problem where the goal was to predict if Player 1 wins or loses.

During Phase 1, we explored the dataset, analyzed the most played cards and studied the correlations between cards and victories. We discovered that the dataset follows the game meta and is highly imbalanced, with around 72% of victories for Player 1. We trained three baseline models and observed that Logistic Regression was the most stable one, while KNN and Decision Tree were strongly affected by the imbalance.

In Phase 2, we enriched the dataset using an external dataset containing elixir cost, usage rate, win rate and rarity of each card. We created new aggregated features to represent the global strength and style of each deck. We then tested more advanced models such as Random Forest, XGBoost and CatBoost. CatBoost achieved the best performance with an accuracy around 0.73.

However, the improvement remains limited. This project shows that deck composition alone is not enough to accurately predict the result of a Clash Royale match. Many important elements of the game are missing, especially player skill, card levels and in-game decisions.

For future work, a more complete dataset including real gameplay data could allow much better predictions. This project allowed us to better understand the limits of machine learning when the available data is incomplete, and the importance of feature quality in predictive modeling.

12 Scientific References

- Dorogush, A. V., Ershov, V., & Gulinostrov, A. (2018). *CatBoost: gradient boosting with categorical features support.* Advances in Neural Information Processing Systems (NeurIPS).

- Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
- He, H., & Garcia, E. A. (2009). *Learning from imbalanced data*. IEEE Transactions on Knowledge and Data Engineering.